MST124

Essential mathematics 1

Computer algebra guide

This publication forms part of an Open University module. Details of this and other Open University modules can be obtained from Student Recruitment, The Open University, PO Box 197, Milton Keynes MK7 6BJ, United Kingdom (tel. +44 (0)300 303 5303; email general-enquiries@open.ac.uk).

Alternatively, you may visit the Open University website at www.open.ac.uk where you can learn more about the wide range of modules and packs offered at all levels by The Open University.

The Open University, Walton Hall, Milton Keynes, MK7 6AA.

First published 2014. Second edition 2019.

Copyright © 2014, 2019 The Open University

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, transmitted or utilised in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without written permission from the publisher or a licence from the Copyright Licensing Agency Limited, 5th Floor, Shackleton House, Hay's Galleria, 4 Battle Bridge Lane, London SE1 2HX (website www.cla.co.uk).

Open University materials may also be made available in electronic formats for use by students of the University. All rights, including copyright and related rights and database rights, in electronic materials and their contents are owned by or licensed to The Open University, or otherwise used by The Open University as permitted by applicable law.

In using electronic materials and their contents you agree that your use will be solely for the purposes of following an Open University course of study or otherwise as licensed by The Open University or its assigns.

Except as permitted above you undertake not to copy, store in any medium (including electronic storage or use in a website), distribute, transmit or retransmit, broadcast, modify or show in public such electronic materials in whole or in part without the prior written consent of The Open University or in accordance with the Copyright, Designs and Patents Act 1988.

Edited, designed and typeset by The Open University, using LaTeX.

Printed in the United Kingdom by Latimer Trend and Company Ltd, Plymouth

SUP 06126 3

2.1

Contents

Introduction **5**

1 Installing, testing and configuring Maxima **7**
1.1 Introducing and installing Maxima 7
1.2 Starting, testing and configuring wxMaxima 9

2 Getting started with Maxima **14**
2.1 Calculating with Maxima 14
2.2 wxMaxima cell markers 16
2.3 Decimal numbers 17
2.4 Troubleshooting wxMaxima 19
2.5 Reusing and editing commands 20
2.6 Variables 22
2.7 System variables 25
2.8 Annotating your wxMaxima worksheet 26
2.9 Saving and printing your work 27
2.10 Getting help 30

3 Algebra **32**
3.1 Algebraic manipulations 32
3.2 Equations and their solutions 34
3.3 Plotting graphs 39

4 Functions **42**

5 Exponential functions and logarithms **44**

6 Trigonometry **47**

7 Plotting circles **54**

8 Differentiation **59**

9 Integration **62**

10 Matrices **68**
10.1 Matrix algebra 68
10.2 Determinants and inverses 74

11 Sequences and series **75**
11.1 Plotting graphs of sequences 75
11.2 Summing series 81

12 Taylor polynomials **82**

13 Complex numbers **84**
13.1 Working with complex numbers 84
13.2 Solving polynomial equations 86

14 Maxima accessibility guide **89**
14.1 Choosing and configuring your interface 89
14.2 Starting, testing and configuring Maxima using the command-line interface 91
14.3 Troubleshooting Maxima using the command-line interface 95
14.4 Reusing and editing commands in the command-line interface 95
14.5 Saving and printing your work using the command-line interface 96
14.6 Plotting graphs without wxMaxima 100
14.7 Changing properties of graphs 100

Computer methods for CAS Activities in Books A–D **102**

Solutions to Computer activities **114**

Maxima reference guide **155**

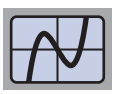
Acknowledgements **162**

Index **163**

Introduction

Throughout this module you will use computer software called **Maxima** to investigate mathematical concepts and perform calculations that may be too laborious or difficult to do by hand. Maxima is a **computer algebra system** (often abbreviated to **CAS**); that is, a computer program that can perform algebraic manipulations, such as expanding brackets, factorising expressions and solving equations, as well as performing numerical calculations and plotting graphs. It is typical of the type of computer software used by mathematicians today.

This *Guide* explains how to use Maxima for some of the mathematics in MST124. It is designed to be read in conjunction with the main study texts and is not meant to be read all at once because its later sections use mathematical concepts that you may not have yet met. As you work through the main study texts in Books A–D, you will come across sections or activities marked with the icon shown here. Some of these ask you to study part of this *Guide*. Other activities ask you to solve a problem using Maxima commands that you have previously learned. While the mathematical answers to such activities are included within the appropriate book, the computer methods needed to solve them are given in the section 'Computer methods for CAS activities in Books A–D' towards the end of this *Guide*.



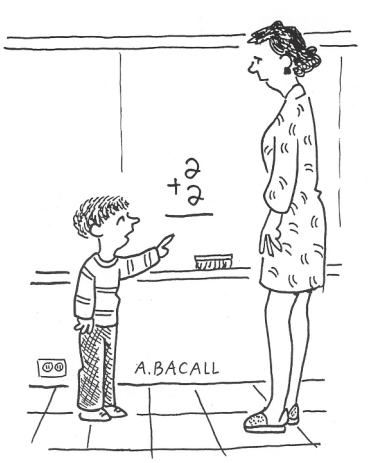
The 'Maxima accessibility guide' (Section 14) is primarily aimed at those who may have difficulty using Maxima because of a disability, and provides some advice and suggestions.

The final section of this *Guide* contains a summary of all the Maxima commands introduced. This should be useful for reference, both as you work through the module and afterwards.

Note that this *Guide* does not attempt to describe all the features of Maxima. If you need more information, please consult the extensive documentation provided on the Maxima website at `maxima.sourceforge.net` or use Maxima's help system, which is described in Subsection 2.10.

Maxima's origins lie in a system called Macsyma, a well-respected computer algebra system first developed in the late 1960s at the Massachusetts Institute of Technology and funded by the United States Department of Energy. From 1982 until his sudden death in 2001, William (Bill) Schelter, a professor of mathematics at The University of Texas at Austin, maintained a version of Macsyma which we now know as Maxima. In 1998 he obtained permission from the Department of Energy for the system to be made freely available to all. Maxima is now dedicated to his memory.

William Schelter (1947–2001)



"Rather than learning how to solve that, shouldn't I be learning how to operate software that can solve that?"

A word of warning

Using a computer algebra system is not a substitute for learning the mathematical techniques taught in this module! You need to become proficient at using the methods taught so that you can apply them in the examination and in your future studies, and also to help you develop your mathematical understanding and intuition.

Maxima is not intelligent – it is just a tool, similar to your calculator. You need to use your own mathematical skills to guide it and interpret its results critically.

Conventions

Here are the various computing terms used in this *Guide*, and how they should be interpreted on Microsoft Windows and Apple Mac computers. The *Guide* concentrates on using Maxima on a Microsoft Windows computer, although Maxima is available for many different types of computer.

Computer terminology

Term	Microsoft Windows	Apple Mac
Click	Click the left mouse/touchpad button	Click the mouse/touchpad button
Right click	Click the right mouse/touchpad button	Hold the Ctrl key while clicking the mouse/touchpad button
Select	*Either* click to select an option, *or* select text by dragging the cursor over the text while pressing the left mouse button	*Either* click to select an option, *or* select text by dragging the cursor over the text while pressing the mouse button
`Ctrl-Q` Etc.	Hold down Ctrl while pressing Q Etc.	Hold down Command while pressing Q Etc. – but there are exceptions to this pattern that we'll mention as they arise

On some computer keyboards the Enter key might be labelled Return, or simply ↵, and the Shift key might be labelled ⇧.

In this *Guide*, these keys are referred to as Enter and Shift, respectively.

Sometimes Enter and Return are two separate keys. For such keyboards references in this *Guide* to the Enter key should be interpreted as references to the Return key.

1 Installing, testing and configuring Maxima

1.1 Introducing and installing Maxima

Maxima can be used on many different types of computer, and there are several different intermediary systems, known as **interfaces** or **front-ends**, that provide a means of communication between you, the user, and the Maxima system itself. This is illustrated in Figure 1. The interfaces enable you to enter commands that are then processed by Maxima, and they display the results obtained.

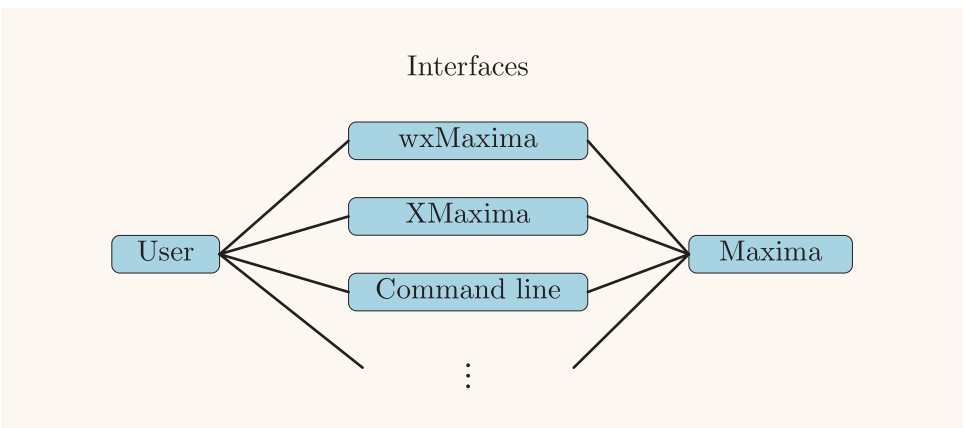


Figure 1 Different interfaces for Maxima

All the commands mentioned in this *Guide* can be used with any interface to Maxima, unless indicated otherwise.

The recommended interface is **wxMaxima**, which is shown in Figure 2. (The letters 'wx' in the name of this interface refer to the software used to create it.) If you need to use an interface other than wxMaxima, for example if you are not using a Windows computer and you are unable to install wxMaxima, then you should try to achieve all the outcomes shown in this *Guide* using your chosen interface.

If you use a screenreader, then the basic **command-line** interface shown in Figure 3 may be the easiest interface for you to use.

If you think you may have difficulty using the wxMaxima interface, because of a disability for example, then you should read Subsection 14.1 now. This contains details of how wxMaxima might be configured to meet your needs, and should help you decide whether to use wxMaxima or the command-line interface.

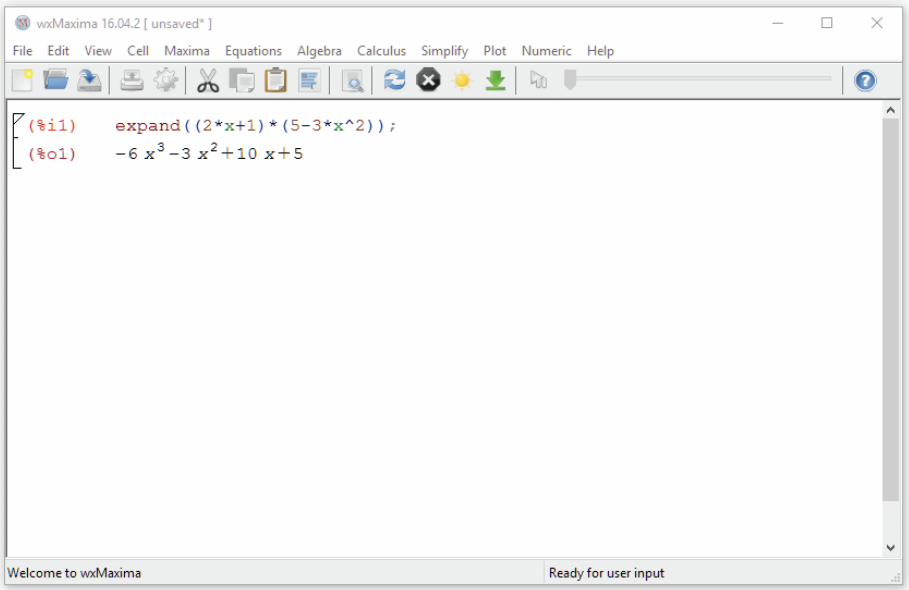


Figure 2 The wxMaxima interface

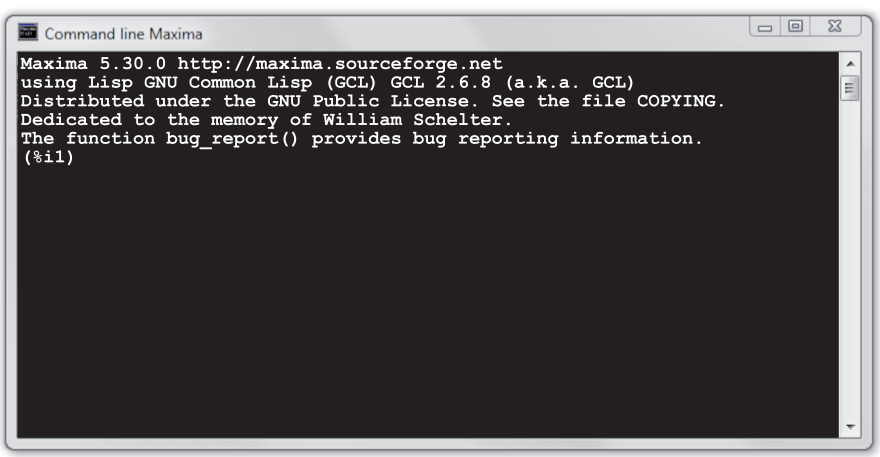


Figure 3 The command-line interface to Maxima

Before you can use Maxima, you need to install it on your computer. You will do this in the next activity.

Computer activity 1 Installing Maxima

(a) Go to the OU Maxima website at `learn1.open.ac.uk/site/maxima` and follow the instructions there to download and install Maxima on your computer. (You can also find a link to the OU Maxima website on the MST124 website.)

(b) Create a folder on your computer in which to store your Maxima work for this module. You might like to put this on your computer desktop, or in your usual documents area.

If you are studying within a prison or other closed institution, you can install Maxima from the *Offline resources* disc.

1.2 Starting, testing and configuring wxMaxima

If you are using the command-line interface you should study Subsection 14.2 instead of this subsection. If you are using a different interface, you should make sure that you can achieve similar outcomes with that interface.

Computer activity 2 Starting wxMaxima

Start wxMaxima on your computer, and keep it open to work with as you study this section.

If you cannot remember how to do this on your computer, see the OU Maxima website: `learn1.open.ac.uk/site/maxima`.

The wxMaxima interface to Maxima, illustrated in Figure 4, consists mainly of a large white working area (called the **worksheet**) where you can type Maxima commands. These commands instruct Maxima to either perform calculations or change system settings.

When the software first starts, a secondary window showing the 'Tip of the Day' is also displayed. You can close it by clicking the 'Close' button. If you want to prevent this window from appearing the next time you start wxMaxima, click the box in its bottom left-hand corner, so that it is no longer ticked, before clicking 'Close'. (You can always reinstate these messages at any time by selecting `Show tips` from the `Help` menu.)

Along the top of the main window (or along the top of the screen in the case of an Apple Mac) is a list of menu names, including the `File` menu which allows you to save or print your work and re-open previously saved sessions (there is more on this later). You can also close the software from the `File` menu by selecting `Exit`. (On an Apple Mac you can do this by selecting `Quit wxMaxima` from the `wxMaxima` menu.)

In Figure 4, a row of icons known as the **toolbar** is shown below the row of menu names. This is displayed only if `Main Toolbar` is selected in the `View` menu. The icons provide shortcuts to commonly used functions.

The `View` menu also allows you to display palettes of commonly used commands. You can use these palettes as an alternative to typing some of the commands given in this *Guide*.

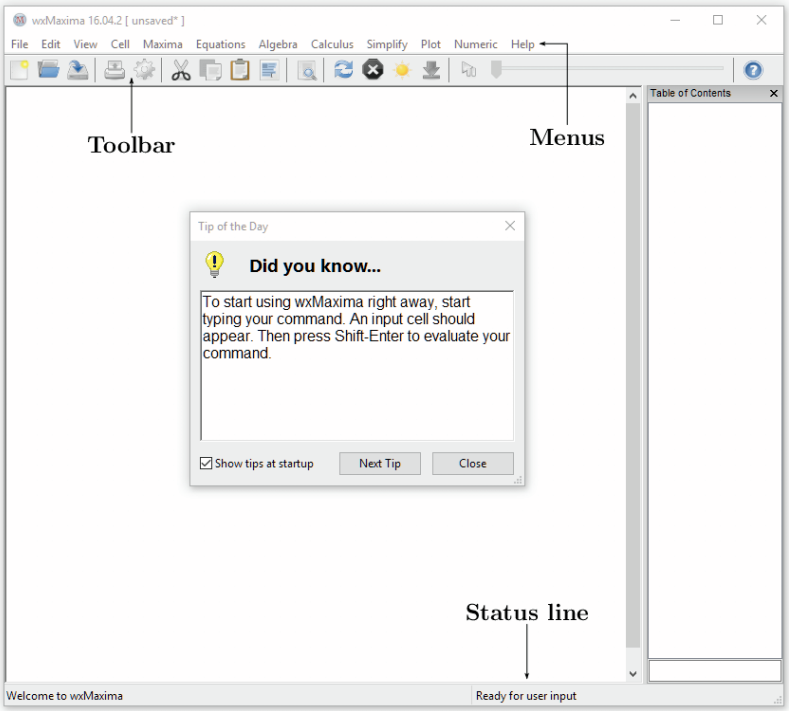


Figure 4 The initial wxMaxima window

Also shown in Figure 4 is the **Table of Contents**. This shows the names of any sections or subsections that you may add to your worksheet (see Subsection 2.8). You may want to close the Table of Contents, by clicking the cross near its top right-hand corner.

At the bottom right-hand corner of the wxMaxima window is a message that indicates the current status of the Maxima system. When Maxima is waiting for instructions from you this reads 'Ready for user input'. Other possible messages include 'Maxima is calculating', or 'Reading Maxima output'.

The following activity shows you how to use wxMaxima to perform a simple calculation and checks that your system is working properly. Follow the steps using wxMaxima yourself as you read the activity.

If you encounter unexpected problems when working through this activity, check the Frequently asked questions (FAQs) section on the OU Maxima website: `learn1.open.ac.uk/site/maxima`.

Computer activity 3 Your first Maxima calculation

Follow these steps to use Maxima to calculate $2 + 3 \times \frac{4}{5}$.

(a) Click anywhere on the large white area of the wxMaxima window, to ensure that the text you type next is directed to the correct place.

(b) Type the following exactly as it appears here:

```
2+3*4/5;
```

including the semicolon (`;`) at the end.

Notice that as you type your command, it is shown in wxMaxima preceded by a red arrow: `-->`.

This is the **input prompt** and shows where commands can be typed.

To the left of the input prompt is the **cell marker**

which will be described in more detail later.

As you type, the colour of the text changes to highlight the different elements of your expression: numbers are coloured differently to the operations.

(c) Press `Shift-Enter`, that is, hold down the Shift key while pressing Enter.

(Remember: if, as on some Macs, your keyboard has separate Enter and Return keys, then you should hold down the Shift key while pressing Return.)

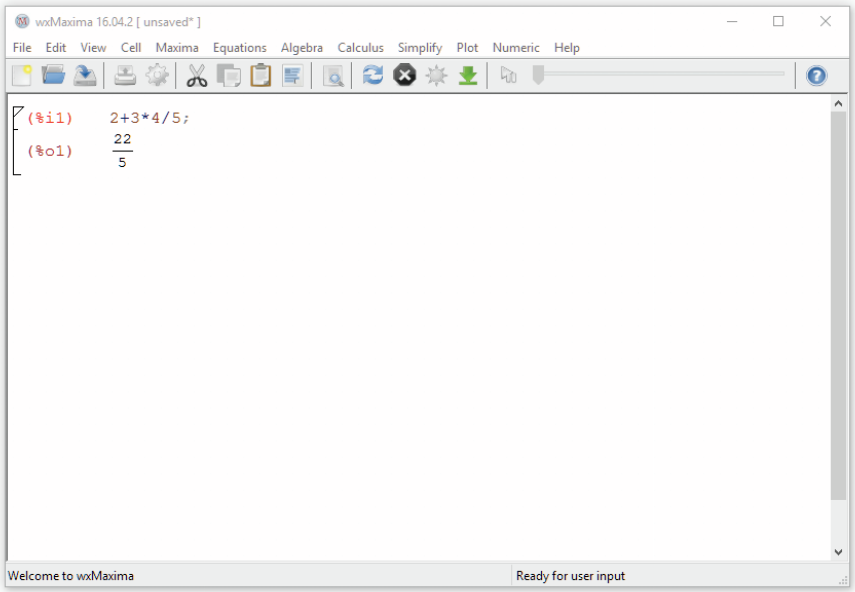
If you make a mistake, try clicking elsewhere on the worksheet, below your error, and try again.

Pressing `Shift-Enter` instructs wxMaxima to send the expression to Maxima to calculate. This is known as **evaluating** the expression.

The input prompt (`-->`) is now replaced by `(%i1)`. This is how Maxima labels lines. The `i` stands for 'input': this is input line number 1. Within each Maxima session line numbers start at 1 and increase for each new input line. The percentage symbol (`%`) is used by Maxima to indicate objects built into the system. There is more about this later.

The status message in the bottom right-hand corner of the wxMaxima window first changes to 'Maxima is calculating', then 'Reading Maxima output', before the result of the calculation is displayed. The message then displays 'Ready for user input' when the system is ready to accept another command.

The result is displayed under the input line, preceded by the label `(%o1)`. The `o` stands for 'output', so this label identifies the line as output line 1. It gives the output corresponding to input line 1.



If, in Computer activity 3, Maxima behaved as described, then you have a working system to use throughout MST124. Before finishing though, you should make a couple of small changes to how the wxMaxima interface behaves.

In Computer activity 3 you used `Shift-Enter` rather than Enter to instruct Maxima to evaluate expressions, and the input prompt (-->) appeared only once you started typing. Both these behaviours can be disconcerting, but can be changed, as described in Computer activity 4.

It is recommended that you configure wxMaxima as described in Computer activity 4, and in the remainder of this *Guide* it is assumed that you have done so.

Computer activity 4 Configuring wxMaxima

(a) From the `Edit` menu, select `Configure` (or on an Apple Mac computer, from the `wxMaxima` menu, select `Preferences`).

Alternatively, click the following icon on the toolbar.

This opens the 'wxMaxima configuration' window.

(b) Within the Configuration window, click the icon labelled 'Worksheet' to access the worksheet settings, then tick the 'Enter evaluates cells' box by clicking it.

This changes the behaviour so that pressing Enter alone evaluates commands.

(c) In the same list of settings, tick the 'Open a cell when Maxima expects input' box by clicking it.

This ensures that the input prompt is shown whenever Maxima is waiting for input.

(d) In the same list of settings, ensure the 'Show user-defined labels instead of (%oxx)' box is *not* ticked.

This ensures that the output line numbers are always shown.

(e) Click on the icon labelled 'Options' in the Configuration window, then tick the 'Save sidebar status' box by clicking it.

This ensures that your preferred configuration of the wxMaxima window, for example hiding the Table of Contents, is retained between Maxima sessions.

(f) Click 'OK' (or close the window if using an Apple Mac).

Closing wxMaxima

When you have finished using Maxima, you can close Maxima and the wxMaxima interface by using one of the following methods.

- Select `Exit` from the `File` menu (or `Quit wxMaxima` from the `wxMaxima` menu on an Apple Mac computer).
- Press `Ctrl-Q`. (Here, `Q` stands for quit.)
- On a Windows computer, click the usual small cross button at the top right-hand corner of the wxMaxima window.

You may be asked if you want to save your changes to the worksheet before closing. At this stage, click 'No'. You will see how to save your work in Subsection 2.9.

2 Getting started with Maxima

In this section you will learn about the basics of the wxMaxima interface and how to use it to perform simple calculations with Maxima.

If you encounter unexpected problems when working through this section, remember to check the Frequently asked questions (FAQs) section on the OU Maxima website.

2.1 Calculating with Maxima

In Computer activity 3 you were asked to enter a numerical expression into Maxima using the symbols `+` for addition, `*` for multiplication and `/` for division. Maxima evaluated it using the rules for the correct order of mathematical operations; that is, the BIDMAS rules. The way in which a calculation or a command has to be entered in Maxima is known as its **syntax**. The Maxima syntax for the basic mathematical operations and other functions is summarised below.

In this *Guide* we will use ^ for powers. Many Maxima commands take the form of a command name, such as `sqrt` or `abs`, followed by a pair of round brackets containing one or more objects, such as numbers, that the command operates on. Such an object is called an **argument** of the command. In the summary tables in this *Guide*, such as the one below, arguments of commands are often represented by ▯.

Basic Mathematical operations

Operation or function	Syntax	Example
Addition	`+`	`4+3;`
Subtraction	`-`	`4-3;`
Multiplication	`*`	`4*3;`
Division	`/`	`4/3;`
Brackets	`(` *and* `)`	`2*(3+4);`
Powers, for example 2^3	`^`	`2^3;`
	or `**`	`2**3;`
Square root, for example $\sqrt{5}$	`sqrt(`▯`)`	`sqrt(5);`
Magnitude (absolute value)	`abs(`▯`)`	`abs(-3);`

You should end all Maxima commands with either a semicolon (`;`) or a dollar sign (`$`). Ending a command with a dollar sign tells Maxima to perform the instruction, but not to display the answer. When using wxMaxima, if you forget to finish your command with a semicolon or dollar sign, then the system will add a semicolon for you. Other interfaces may not do this. In such systems, pressing Enter without ending the line with a semicolon or dollar sign simply moves the editing cursor to the next line, waiting for you to finish entering a complete command.

You can also enter several commands on one line, ending each one with a semicolon or dollar sign.

Maxima generally ignores all spaces that you type while entering a calculation or command, so you can use spaces to make a command easier to read. Once wxMaxima has been configured to use Enter to evaluate commands (as described in Computer activity 4), you can then use `Shift-Enter` to enter line breaks, which are also ignored by Maxima.

Warning: Do not omit multiplication signs!

A common error when first using Maxima is to omit multiplication signs. For example, you might type

- `2sqrt(2)` rather than `2*sqrt(2)`, or
- `2(3+4)` rather than `2*(3+4)`.

Doing this will result in an error. The displayed error message might include one of the following statements:

- *parser: incomplete number; missing exponent?*
- *incorrect syntax: Syntax error*
  incorrect syntax: Too many)'s
  incorrect syntax: Premature termination of input at ;.
- *... is not an infix operator.*

If you receive one of these error messages, first check for missing multiplication signs!

Warning: Take care with question marks (?)

Do not type question marks when asking Maxima to evaluate an expression. The symbol `?` is a special symbol in Maxima which provides access to underlying systems.

The one exception to this is when you are using the commands for obtaining help from the system, which you will see in Subsection 2.10.

You can calculate roots other than square roots, such as cube roots, by using the index laws.

Computer activity 5 Calculating cube roots

(a) Remembering that $\sqrt[3]{27} = 27^{\frac{1}{3}}$, calculate $\sqrt[3]{27}$ by entering the following line into Maxima.

```
27^(1/3);
```

Remember to press Enter to evaluate the command.

(b) What happens if you enter

```
27^1/3;
```

instead? Can you explain this?

During the previous activity you may have noticed that when you enter an opening bracket, the wxMaxima interface automatically adds a closing one for you. This behaviour can be turned off by unticking 'Match parenthesis in text controls' in the 'Worksheet' settings of the Configuration window. (See Computer activity 4 for how to open the Configuration window.)

2.2 wxMaxima cell markers

Corresponding matching input and output lines in wxMaxima form a **cell** identified with the marker shown in Figure 5.

Figure 5 The wxMaxima cell marker

The top part of the cell marker (between the upper two horizontal marks) corresponds to the input line(s), and the bottom part corresponds to the output line(s). The cell marker turns red while an input line is being edited, and is shown surrounded by a box while the cell is being evaluated, as illustrated in Figure 6.

Figure 6 The cell marker during evaluation

Computer activity 6 Using cell markers

(a) Click on the triangle at the top of the cell marker.

This collapses the cell to a single line. This can be useful if a calculation or result is very long and so makes the worksheet difficult to read.

The triangle at the top of the cell marker becomes filled.

(b) Click the filled triangle at the top of the cell marker.

This expands the cell again.

(c) Click part of the cell marker other than the top triangle to highlight it, then press the Delete keyboard key.

This deletes the cell from the worksheet.

2.3 Decimal numbers

Maxima usually tries to calculate quantities *exactly* rather than approximating them by decimal numbers. It does this by manipulating the symbols in the expression according to the rules of mathematics. This is sometimes known as **symbolic computation**.

If you wish to have a result displayed as a decimal number, you can instruct Maxima to do so with the `float` command. The name of the command `float` arises from the fact that the method used by computers to store decimal numbers internally using the binary digits 0 and 1 is called a *floating-point representation.* The `float` command instructs Maxima to convert a number stored symbolically as a mathematical expression to one stored using a floating-point representation.

Computer activity 7 Obtaining exact and approximate answers

In Maxima, calculate 1309/3451 in three different ways, by entering each of the following expressions in turn.

(a) `1309/3451;` (b) `1309.0/3451;` (c) `float(1309/3451);`

Maxima returns an exact answer for part (a).

Since part (b) involves a decimal number, 1309.0, Maxima returns a decimal approximation to the answer.

In part (c), the command `float` instructs Maxima to return a decimal answer.

Converting to decimal numbers

Operation	Command	Example
Convert to a decimal number	`float( )`	`float(sqrt(2));`

Maxima usually displays decimal numbers to 16 significant figures (though it suppresses trailing zeros at the end of the decimal part of a number; for example 0.125 is displayed simply as `0.125` rather than as `0.1250000000000000`). This is also the precision to which Maxima performs decimal calculations. You will see in Subsection 2.7 how to change the number of significant figures displayed.

You may also see decimal numbers displayed in the form

`6.223015277861141 10`^{139}

which is wxMaxima's way of displaying scientific notation. It means $6.223\,015\,277\,861\,141 \times 10^{139}$. (The command-line interface would display this as `6.223015277861141E+139`.) You can also enter numbers given in scientific notation. For example, to enter 6.022×10^{23}, you can type

```
6.022e23
```

Computer activity 8 Simplifying and approximating numerical expressions

For each of the following expressions, use Maxima to simplify it and then find a decimal approximation for it.

(a) $\sqrt{147}$ (b) $\left(\frac{8}{5}\right)^{2/3}$ (c) 2^{500}

The expression in part (c) doesn't involve decimal numbers and so the result is evaluated exactly. Notice, however, that the middle 91 digits of the answer are not displayed because the system does not expect you to read them all!

The decimal form of the result, `3.273390607896142 10`^{150}, is given in wxMaxima's way of displaying scientific notation. It means $3.273\,390\,607\,896\,142 \times 10^{150}$.

Maxima knows the values of standard mathematical constants such as the irrational numbers e and π. It treats them symbolically unless you explicitly ask for them in floating-point form.

The constants e and π

Constant	Syntax	Example
e	`%e`	`%e^2;`
π	`%pi`	`2*%pi;`

As with the input and output line labels, the '`%`' symbol indicates the name of a quantity built into Maxima.

When displaying mathematics in an output line, wxMaxima displays `%pi` as π.

Computer activity 9 Working with built-in constants

Use Maxima to calculate the value of the expression $\pi + \frac{\pi}{3} + e^2$, both exactly and as a decimal number.

Notice that in the exact answer, `%pi` is displayed as π, but `%e` is unchanged.

2.4 Troubleshooting wxMaxima

If you are using wxMaxima and it does not seem to be responding, try the following suggestions.

(Suggestions for troubleshooting Maxima when using the command-line interface are given in Subsection 14.3.)

Troubleshooting wxMaxima

Cell Marker	Status message	Comments
Boxed:	Maxima is calculating	Maxima is evaluating a command that is taking a long time to complete.
Boxed:	Ready for user input	Maxima is waiting for you to type something before continuing. In either case you might wish to abort the command in one of the following ways. • Click on the toolbar. • Select `Interrupt` from the `Maxima` menu. • Type `Ctrl-G` (on an Apple Mac hold down the Command key while pressing .). Maxima might take some time to respond to this request.
	Maxima process terminated	Maxima (but not the wxMaxima interface) has stopped working. Select `Restart Maxima` from the `Maxima` menu.

If you encounter other problems, check the Frequently asked questions (FAQs) section on the OU Maxima website: `learn1.open.ac.uk/site/maxima`.

2.5 Reusing and editing commands

In Computer activity 8 you may have entered each expression twice: once to evaluate it exactly and once for a decimal answer. This is, however, not necessary. When you have entered and evaluated an expression, you can edit it and then evaluate it again, as shown in the following activity.

(Details on reusing and editing commands when using the command-line interface are given in Subsection 14.4.)

Computer activity 10 Editing a command

(a) Calculate $2\sqrt{3}+5\sqrt{27}$ by entering the following line into Maxima.

```
2*sqrt(3)+5*sqrt(27);
```

(b) Edit your entry in part (a) to calculate $2\sqrt{3}-5\sqrt{27}$ as follows.

(i) Click on the expression you entered in part (a), or move to it using the up and down keyboard arrow keys.

As you do this the cell marker will turn red and you will see a small flashing vertical line (the **editing cursor**) appear in the expression.

```
(%i1) 2*sqrt(3)+5*sqrt(27);
```

(ii) Edit the expression by using the ← (backspace) or Delete key to delete the + sign and enter a - sign instead.

(iii) Press Enter.

Pressing Enter re-evaluates the cell. When a cell is re-evaluated in this way, the input and output line numbers are updated as if a new cell had been added.

Should you click in the space *between* two cells, a horizontal cursor will be shown, as in Figure 7. This indicates the position at which a new cell will be created, if you begin typing here. This is also shown as you move between cells using the up and down keyboard arrow keys.

```
(%i1) %pi+%pi/3+%e^2;
(%o1)  4π/3 + %e^2
(%i2) float(%pi+%pi/3+%e^2);
(%o2) 11.57784630371704
```

Figure 7 The wxMaxima horizontal cursor, positioned between two cells

In wxMaxima you can also copy and paste expressions, or parts of expressions, as you might copy and paste when using other software; that is, using `Copy` and `Paste` from the `Edit` menu, or the `Ctrl-C` and `Ctrl-V` keyboard shortcuts.

Maxima also permits the result of one calculation to be used in another calculation. The symbol `%` represents the result of the last command evaluated. Due to the updating of line numbers when a command is re-evaluated, this might not be the last command in the worksheet, but it will be the command with the highest line number. So, for example, if you enter `sqrt(%);` Maxima calculates the square root of the last evaluated output. You can also use the output (or input) of other lines, by typing the line label in a calculation. For example, if you enter `%o5^3;` Maxima calculates the cube of the expression in output line 5.

Computer activity 11 Using and editing previous calculations with wxMaxima

(a) Enter `6+2;` into a cell at the bottom of your worksheet and evaluate it.

(b) On the following line type `3+%;` and enter it.

> Maxima returns 11, the sum of 3 and the previous answer.

(c) Edit the line where you typed `6+2;`, change it to `6*2;` and re-evaluate the cell by pressing Enter.

> Notice that the cell number and output of the line you edited is updated, but the result of `3+%;` does not change. This is because when the `3+%;` cell was evaluated, the symbol `%` represented the result of the old version of the edited command.

(d) From the `Cell` menu, select the `Evaluate All Visible Cells` option, or press `Ctrl-R`.

> This re-evaluates all visible cells in the worksheet, *in the order in which they appear*, updating all the line numbers as it does so.
>
> Notice that the result of `3+%;` has now changed. At the time it was re-evaluated the symbol `%` represented the value of the last evaluation, which was the revised version of the previous line.

(e) In a new cell at the bottom of your worksheet enter `%o  +5;` with ▯ replaced by the output line number corresponding to the result of `6*2` entered earlier.

This command adds 5 to the result of the line referenced.

Re-evaluating a wxMaxima worksheet

Selecting the `Evaluate All Visible Cells` option from the `Cell` menu re-evaluates all visible cells in the worksheet, in the order in which they appear.

This can also be achieved by pressing `Ctrl-R`.

By default, the re-evaluation of a worksheet will stop if an error is encountered. This behaviour can be changed so that evaluation continues past the error by unticking 'Abort evaluation on error' in the 'Maxima' settings of the Configuration window.

2.6 Variables

Maxima allows you to assign a value to a *variable*, and then use that variable in further calculations.

For example, you can assign the value 23 to the variable `a`, and then use `a` in subsequent expressions or commands, such as $a^2 - 3a + 2$. You can also assign expressions, both numerical and (as you will see later) algebraic, and indeed any other Maxima object, to a variable, and use that variable in subsequent commands.

To assign a value or expression to a variable you use a colon (`:`). For example, the command

```
a:23;
```

assigns the value 23 to the variable `a`. The colon can be read as 'is assigned the value'.

Warning

You cannot use `=` to assign a value to a variable. This symbol is used to define *equations* in Maxima.

Variable names can be any combination of letters and numbers that begins with a letter. For example, Maxima will accept any of the following variable names.

```
a    A    solution    solution2    x2b
```

Some words are built-in to Maxima and cannot be used as variable names. If you try to assign such a variable, you will get an error such as:

```
assignment:  cannot assign to ...
-- an error.  To debug this try:  debugmode(true);
```

If you encounter such an error, choose a different name for your variable.

Variable names are *case-sensitive*; for example, the variable `x` is different from the variable `X`. If a variable name is the name of a Greek letter, then wxMaxima will display it in output lines as the proper Greek character; for example, the variable `alpha` will be displayed as α.

Once a variable has been assigned a value, Maxima will remember this value for the rest of your session. This may sometimes surprise you. It is easy to forget that earlier in the day you assigned a value to, say, `x`; you may then be confused later on when a calculation involving `x` gives an unexpected result. You can, however, tell Maxima to 'forget' about a variable to which you have previously assigned a value. You can also ask it to list all the variables that you have assigned values to. These commands are demonstrated in the following activity.

Computer activity 12 Working with variables

(a) Assign the value of $\sqrt{8}$ to the variable `a` by entering

```
a:sqrt(8);
```

The value of `a` is displayed as output, in an equivalent form.

Note that when assigning variables, wxMaxima precedes the output with either the output line number or the name of the variable being assigned a value. Which is shown is determined by whether or not 'Show user-defined labels instead of (%oxx)' is ticked in the 'Worksheet' settings of the Configuration window. In this *Guide*, the output line numbers are shown.

(b) Enter `a;`

This displays the current value of `a`.

(c) Assign the value of $\texttt{a}\sqrt{2}$ to `b`.

Don't forget to include a multiplication sign when inputting $\texttt{a}\sqrt{2}$.

The value of `b` is simplified when displayed.

(d) Edit the line where you assigned $\sqrt{8}$ to `a`, so that the value $\sqrt{7}$ becomes assigned to `a`.

(e) Enter `b;` to display the value of `b`.

Notice that the value of `b` has not changed. The variable `b` was defined when `a` had its original value.

(f) Enter `values;` to list the names of all the variables that are currently assigned values.

The variable names are given within square brackets, separated by commas. This is how Maxima displays *lists*.

(g) Enter `kill(a);` to remove the variable `a` from the system.

You should obtain the output *`done`*.

(h) Enter `a;` to display the value of `a`.

The variable name is output, as `a` no longer has a value.

(i) Enter `b;` to display the value of `b`.

`b` is still defined.

Notice that in Computer activity 12(d) and (e), when the value of `a` was changed, the value of `b` was not affected, since it was defined when `a` had its original value. If you want to update the value of `b` using a different value of `a`, then you should enter a new value of `a` then re-evaluate the line defining `b`. You can do this in wxMaxima by selecting the line and pressing Enter, or re-evaluating the whole worksheet by selecting the option `Evaluate All Visible Cells` from the `Cell` menu.

Working with variables

Operation	Command	Example
Assign a value to a variable	`variable:value`	`a:23;`
Display the value of a variable	`variable`	`a;`
List all user-assigned variables	`values`	`values;`
Remove an assigned variable	`kill(variable)`	`kill(a);`
Remove all assigned variables	`kill(all)`	`kill(all);`

Note: here the placeholder `variable` represents any variable name.

You can practise using variables in the following activity.

Computer activity 13 Using variables

(a) Define the variable `a` to have the value 42.

(b) Define the variable `b` to be equal to $\texttt{a}^3 - 3\texttt{a} + 4$.

(c) Define the variable `c` to be the decimal approximation of the square root of `b`.

2.7 System variables

There are some variables built into Maxima whose values affect the behaviour of the system. These are called **system variables**. You met one system variable in Computer activity 12: the variable `values` holds a list of the names of all the variables you have defined.

Another system variable, `fpprintprec`, specifies the number of significant figures of decimal numbers that are displayed. The name `fpprintprec` is an abbreviation of 'floating-point print precision'.

To change the system behaviour so that, for example, only 4 significant figures are displayed, use the command `fpprintprec:4;` to assign the value 4 to the variable `fpprintprec`.

You can set the variable `fpprintprec` to any value between 2 and 16. Also, setting it to 0 restores the default behaviour of displaying 16 significant figures. The value of `fpprintprec` sets the *requested* number of significant figures, but Maxima does not always exactly meet the request.

Note that setting `fpprintprec` only affects how decimal numbers are displayed: it does not change the accuracy with which calculations are performed.

Computer activity 14 Displaying a specified number of significant figures

Display the decimal approximation of π to 8 significant figures by doing the following.

(a) Set the system variable `fpprintprec` to be 8.

(b) Display the decimal approximation of π.

You can reset the values of `fpprintprec` and all other system variables to their original values by using the `reset()` command, which has no arguments. Try entering `reset();` followed by `float(%pi);` and check that the value of π is displayed to 16 significant figures. Note that the `reset` command also resets your Maxima line numbers to start at 1 again.

Resetting system variables

Operation	Command	Example
Reset all system variables	`reset()`	`reset();`

Note that the list of variables displayed when you enter `values;` does not include any system variables. If one of your variables does not appear in the list displayed by `values;`, then you have probably used a variable name that is also the name of a system variable and hence changed the value of that system variable. In extreme circumstances this may change the behaviour of Maxima, which can be restored by resetting all system variables as described above.

2.8 Annotating your wxMaxima worksheet

It is important to be able to clearly explain and present your mathematical work, both so that someone else can understand it, and so you can understand what you have done if you return to your work after some time. To help you do this, wxMaxima allows you to enter text comments within your worksheet, as illustrated in Figure 8.

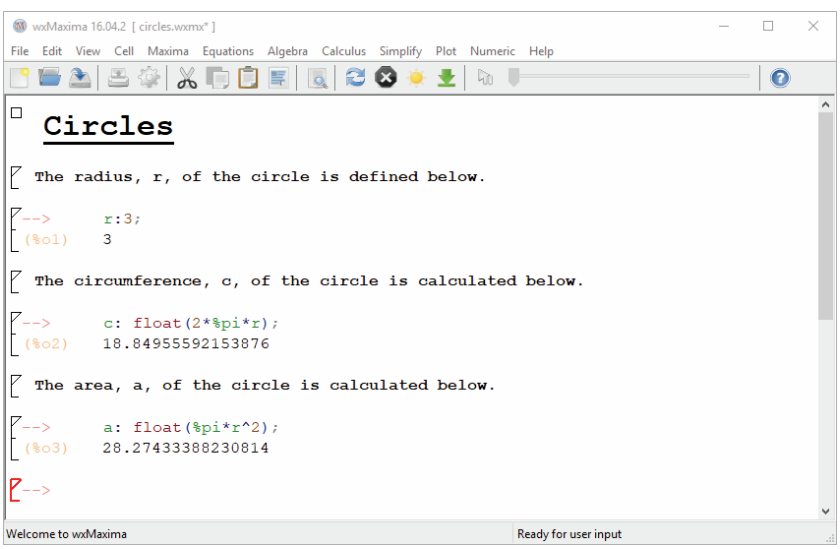


Figure 8 Adding text to a wxMaxima worksheet

To enter text into your worksheet, first you need to create a suitable cell in which to type. To do this, position the cursor and then select the appropriate one of the following commands from the `Cell` menu.

- `Insert Text cell`
- `Insert Title cell`
- `Insert Section cell`
- `Insert Subsection cell`

The new cell will be created under the cell in which the editing cursor appears, or between two existing cells at the position indicated by the

horizontal cursor, if shown. Remember, the horizontal cursor appears whenever you click between two cells, or move within the worksheet using the up and down keyboard arrow keys.

As you type your text, to start a new line within the cell press Enter. When you have finished typing your text, click elsewhere on the worksheet or use the up or down keyboard arrow keys to move out of the text cell. If your text cell is at the bottom of the worksheet, and you click or move below it, an input prompt may not be shown until you start typing again.

Section and Subsection cells are automatically numbered.

Each Title cell, Section cell and Subsection cell has a small square in its top left corner. Clicking this square hides (or reveals) the contents of the worksheet from that point onward, until the next cell of the same type.

Whenever you add a Section or Subsection cell, a corresponding entry is added to the Table of Contents window. You can display this window by selecting `Table of Contents` from the `View` menu, or by pressing `Alt-Shift-T`, that is, by holding down the Alt and Shift keys while pressing T.

Computer activity 15 Including text in a worksheet

(a) Create the worksheet shown in Figure 8. Use a Title cell for the title 'Circles' and Text cells for the remaining text. Remember to click elsewhere on the worksheet or move position with the up or down keyboard arrow keys when you have finished entering each piece of text.

After evaluating a Maxima command, you will need to correctly position the horizontal cursor before inserting the next text cell.

(b) Use your worksheet to find the circumference and area of a circle of radius 5. (Remember to re-evaluate the worksheet, for example, by pressing `Ctrl-R`, after changing the value of `r`.)

2.9 Saving and printing your work

(Details on saving and printing your work when using the command-line interface are given in Subsection 14.5.)

After working with Maxima you will probably want to save your calculations so that you can reuse or read them at a later date. You can do this in wxMaxima using the `Save` or `Save As` options from the `File` menu.

There are various different formats in which you can save your work.

It is recommended that you save your wxMaxima work as `The input without images` with the `.wxm` file type. This saves all your input

but not any calculated output. You have to recalculate the output when you re-open the file, as demonstrated in Computer activity 16.

The alternative file type available is the `Whole document` with the `.wxmx` file type. This saves all the input and the output. Note that when you load a worksheet previously saved as a `.wxmx` file, although both the saved input and output are displayed, the input has not been evaluated by Maxima, so you cannot immediately use any results of the worksheet in new calculations. To re-evaluate the worksheet, and so recalculate all the results it contains, select `Evaluate All Visible Cells` from the `Cell` menu, or press `Ctrl-R`.

Input commands can also be exported to a `Maxima batch file`, with the `.mac` file type, using the `Export` option of the `File` menu. This is for more experienced users of Maxima.

You can open a saved wxMaxima worksheet by using the `Open` option from the `File` menu, or by double-clicking on the saved file on your computer.

Computer activity 16 Saving your work

Save your current worksheet then reload it, by following these steps.

(a) Select `Save As` from the `File` menu.

(b) In the window that appears:

(i) Choose the folder in which to save your work. You might like to save it in the folder that you created in Computer activity 1.

(ii) Choose a name for the file in which the worksheet will be saved.

(iii) Make sure the file is saved as a `The input without images` with the `.wxm` file extension.

(On a Windows computer, make sure that `The input without images (*.wxm)` is selected from the 'Save as type' drop-down list.)

(iv) Click 'Save'.

(c) Close wxMaxima, by selecting `Exit` from the `File` menu, or pressing `Ctrl-Q`, or clicking the small '`x`' at the top right of the window. (On a Mac select `Quit wxMaxima` from the `wxMaxima` menu or hold down the Command key while pressing Q.)

(d) Open your worksheet in wxMaxima again, by either

(i) double-clicking on the file you just saved, or

(ii) following the steps below:

- Start wxMaxima again.
- Select `Open` from the `File` menu. Maxima will ask if you want to save your just opened, blank worksheet. Click 'No'.
- Find the file containing your work and click 'Open'.

(e) Re-evaluate the worksheet by selecting `Evaluate All Visible Cells` from the `Cell` menu or pressing `Ctrl-R`.

The calculations in the worksheet are evaluated and results displayed.

Printing is best achieved by first exporting the whole worksheet to a HTML document (a web page) by selecting `Export` from the `File` menu, then opening the HTML document using a web browser and printing from that. The worksheet can also be exported as source code for the LaTeX mathematical typesetting system using the same menu items. (Currently, these exports are not perfect representations of the worksheet.)

Alternatively, you can obtain an image of the entire wxMaxima window by taking a 'screenshot'. To do this on a Microsoft Windows computer, first click on the wxMaxima window, then press `Alt-PrintScreen` (that is, hold down the Alt keyboard key while pressing the `PrintScreen` key, which may be labelled PrtScr, Prt Scrn or something similar). This stores the image, which you can then paste into a suitable application. If you are using Windows Vista, Windows 7, 8 or 10, you might like to use the 'Snipping Tool', available by typing 'snip' into the Start menu. To take a screenshot on an Apple Mac computer, press `Cmd-Ctrl-Shift-4` (that is, hold down the Command, Ctrl and ⇧ keyboard keys while pressing the 4 key). Next press the space bar, move the camera pointer over the wxMaxima window and click. This stores the image, which you can then paste into a suitable application. You may also like to use the Grab application from the Utilities folder.

You can also paste individual output lines (or parts of output lines) from the worksheet into, for example, word-processed documents. Select the part of the output that you wish to paste, then from the menu obtained by clicking the right mouse button, select one of the following options.

- `Copy` – This copies the Maxima syntax for the selection. It can be useful if you wish to use the Maxima syntax in another document, or use the copied output as input to another Maxima command.
- `Copy as LaTeX` – This copies a LaTeX representation of the selection that can be pasted into a source file of the LaTeX mathematical typesetting system.
- `Copy as MathML` – This copies the selection as MathML, which is an international standard for the representation of mathematics. The MathML can be pasted into, for example, versions of Word from 2007 onwards, where it will be pasted as mathematics that can be edited using the ribbon maths editor. It can also be imported into LibreOffice documents, by, within LibreOffice, first creating a maths object using the `Insert > Object > Formula` menu options, then importing using `Tools > Import MathML from Clipboard`.

- `Copy as Image` – This copies the selection as an image that can be pasted into another document.

Whole cells can also be cut and pasted using the above options, by selecting the cell marker before clicking the right mouse button. However, when doing this, the LaTeX source contains a significant amount of formatting information that you may not require, and the MathML option for a whole cell does not work in the version of wxMaxima current at the time of writing.

2.10 Getting help

There are several ways in which you can obtain help with using Maxima, beyond this *Guide*. In wxMaxima, you can access the complete Maxima manual by selecting `Maxima Help` from the `Help` menu. This opens the window shown in Figure 9. You can use the Contents, Index and Search tabs to look for the information that you want.

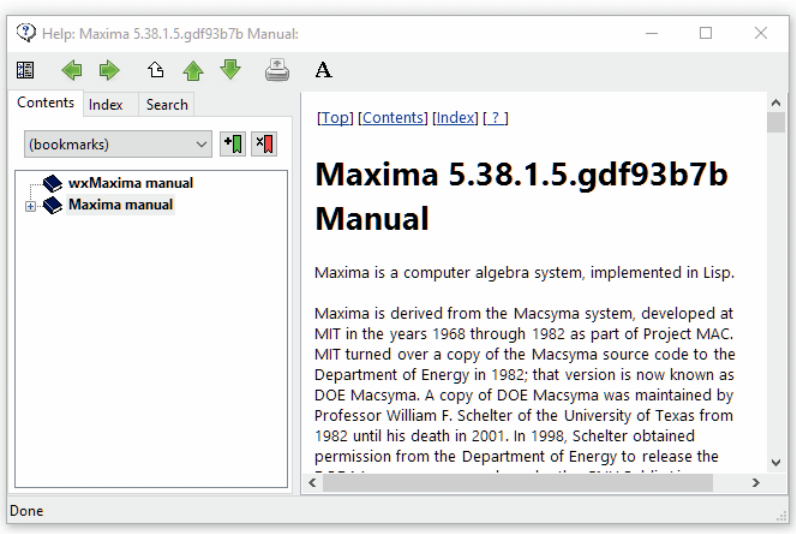


Figure 9 The wxMaxima help system

Alternatively, Maxima also includes commands that you can use to obtain help. These are demonstrated in the following activity and summarised in the table that follows.

Computer activity 17 Getting help!

(a) Enter

```
? float;
```

to show the help for the `float` command.

Note the space between the `?` and `float`, which is needed. This is one of the few times in Maxima when spaces matter!

The '`?`' is the Maxima syntax to request help on the command given.

The help information for the command may include more detail than you need!

After giving the help information, Maxima displays the output `true`. This means that the command you entered was successful.

(b) Suppose that you cannot remember the Maxima command for square root, but you do remember that it was something like *sqr*.

Enter

```
?? sqr;
```

to list all the help information titles that contain the letter sequence 'sqr'.

A list of possible titles is displayed.

Title number 1 is what you were looking for, so enter `1;` to display the relevant information.

If more than one title is displayed in response to a `??` command, then Maxima will not continue until you enter one of the following:

- the appropriate number, to display your chosen information
- several numbers, separated by spaces, to see help information on several topics
- `all` to display the information on all topics listed, or
- `none` to see no information.

A box will be shown around the cell marker while Maxima is waiting for your response.

Help commands

Operation	Command	Example
Get help on a command	`? command` *or* `describe(command)`	`? float;` `describe(float);`
Find help information whose title contains the given text	`??` *or* `describe(, inexact)`	`?? flo;` `describe(flo, inexact);`

Note 1: the space after the `?` is needed.

Note 2: the second pair of commands displays a list of the titles of all help information pages whose title contains the given text. To obtain help on a particular topic listed, type the number corresponding to the required title in the list, followed by a semicolon.

There is further help available in the Frequently asked questions (FAQs) section of the OU Maxima website: `learn1.open.ac.uk/site/maxima`.

3 Algebra

In Section 2 of this *Guide* you used Maxima to perform numerical calculations. Much of the power of Maxima, however, lies in its ability to perform algebraic operations. In this section you will learn how to do this.

3.1 Algebraic manipulations

You can enter algebraic expressions into Maxima in a similar way to numerical expressions, using the symbols `+`, `-`, `*`, `/` and ^, and the command `sqrt( )` for square roots. You can also assign algebraic expressions to variables, using the assign (`:`) command, in the same way as numerical expressions.

In the next activity you will meet some commands for manipulating algebraic expressions.

Computer activity 18 Using algebraic expressions

(a) Enter

```
expand( (2+x)*(3*x+4)^3 );
```

to expand the expression $(2+x)(3x+4)^3$.

> The command `expand` multiplies out the expression that is its argument.
>
> Spaces have been included simply to help with the reading of the command.

(b) Enter

```
factor( 2*x^3-7*x^2-10*x+24 );
```

to factorise the expression $2x^3 - 7x^2 - 10x + 24$.

> The command `factor` factorises, where possible, the expression that is its argument.

(c) Define `p` to be the expression $\dfrac{x^3+1}{x+1}$.

> Remember to use brackets as necessary when typing this expression.

(d) Enter

```
fullratsimp(p);
```

to simplify the expression assigned to the variable `p`.

> The command `fullratsimp` simplifies, where possible, the expression that is its argument. It does this by using techniques for multiplying out brackets, collecting terms over a common denominator and cancelling out common factors. Simplifications requiring other techniques may not be handled by `fullratsimp` in the way you would expect.
>
> The name `fullratsimp` is short for 'full rational simplify'.

(e) Enter

```
subst(3,x,p);
```

to find the value of the expression `p` (defined above) when $x = 3$.

> The command name `subst` is short for 'substitute'.
>
> The command `subst(3,x,p);` substitutes the value 3 for the variable `x` in the expression `p`.

The commands introduced in the previous activity are summarised below.

Algebraic manipulation commands

Operation	Command	Example
Expand brackets	`expand( )`	`expand( (x+1)^2 );`
Factorise	`factor( )`	`factor( 2*x+4*x^2 );`
Simplify	`fullratsimp( )`	`fullratsimp( (2*x+4*x^2)/x );`
Substitute	`subst(value, variable, expression)`	`subst(4, x, x^2+1);` which substitutes `4` for `x` in `x^2+1`

It is hard for a computer to simplify algebraic expressions: often there is no single 'simplest form' for a particular expression, and the computer has no way of knowing which form you prefer. You can use the commands in the table above to simplify an expression in the way you want.

You can access many of these commands from wxMaxima menus. For example, to factorise an expression you can first enter the expression in the worksheet, select it by highlighting the text using the mouse or clicking on the corresponding output line number, then select `Factor Expression` from the `Simplify` menu (or from the menu obtained by right clicking on a highlighted output line number).

Other options on these menus, such as `Substitute...` open a window for you to enter the command options. Options that work in this way are identified by `...` at the end of the option name.

3.2 Equations and their solutions

Maxima can manipulate and solve equations. To type an equation in Maxima, you use an equals sign in the usual way. For example, you can type `3*x+2=4`.

You can assign equations to variables in a similar way to expressions. For example, by entering

```
q: 3*x+2=4;
```

you assign the equation $3x + 2 = 4$ to the variable `q`.

You can extract the left- and right-hand sides of an equation by using the `lhs` and `rhs` commands, as demonstrated in the following activity.

Computer activity 19 Working with equations

(a) Assign the equation $x^2 + 2x = 4$ to the variable `r`.

(b) Extract the left-hand side of the equation by entering

```
lhs(r);
```

(c) Extract the right-hand side of the equation by entering

```
rhs(r);
```

Extracting the left- and right-hand sides of an equation

Operation	Command	Example
Left-hand side of an equation	`lhs(equation)`	`lhs(4*x+1=2*x-2);`
Right-hand side of an equation	`rhs(equation)`	`rhs(4*x+1=2*x-2);`

Note: `equation` indicates that the argument of the command must be an equation, or a variable whose value is an equation.

Maxima can find the exact solution(s) of equations, as demonstrated in the following activity.

Computer activity 20 Solving equations

(a) Solve the equation $x^2 + 2x = 1$, by entering

```
solve( x^2+2*x=1 );
```

Notice that Maxima displays the two solutions of the quadratic equation within square brackets and separated with a comma:

`[x=-sqrt(2)-1, x=sqrt(2)-1]`

This is a Maxima **list**.

(b) Use Maxima to solve $4x^2 - 12x + 9 = 0$.

Notice that although there is only one solution of this quadratic equation, Maxima still gives it as a list, containing just one element.

(c) Use Maxima to solve $2x^3 + x^2 - 5x + 2 = 0$.

This is a *cubic* equation, since the highest power of x is 3. Each cubic equation has at most three solutions.

(d) Try to use Maxima to solve $x^9 + 2x - 4 = 0$.

Here, Maxima returns (within a list) the equation, slightly rearranged, but unsolved.

The `solve` command always tries to find an exact solution of the equation using algebraic manipulations. If it is unable to do this, then it returns the original equation, as here.

In Section 6 of this *Guide* you will see how to find approximate solutions of such equations.

(e) Use Maxima to solve $x^2 - 4x + 5 = 0$.

This equation has no *real* solutions, since the discriminant, $(-4)^2 - 4 \times 1 \times 5 = -4$, is negative.

However, Maxima finds solutions that are *complex numbers*. These involve the imaginary number, usually denoted by i, whose square is -1. Maxima denotes this number by `%i`.

Complex numbers were mentioned in Unit 1 and you will learn more about them in Unit 12.

The `solve` command is summarised below.

Solving an equation

Operation	Command	Example
Solve an equation, exactly	`solve(equation)`	`solve(2*x^2-1=0);`

You saw in Computer activity 20 that the output of the `solve` command is a *list*. In Maxima, a **list** is a collection of individual **elements** separated by commas and enclosed within square brackets. You can learn more about using lists in the following activity.

Computer activity 21 Working with lists

(a) Assign the list containing the first five prime numbers to the variable `A` by entering

```
A: [2,3,5,7,11];
```

(b) Display the fourth element of the list by entering

```
A[4];
```

> The position number of an element in a list is known as its **index**.
>
> You can obtain individual elements of a list by typing the index of the element in square brackets after the list, or after the name of a variable to which the list is assigned.

(c) (i) Assign the solutions of the equation $x^2 - x - 1 = 0$ to the variable `solns`.

(ii) Display decimal approximations to the answers by entering the command

```
float(solns);
```

(iii) Display the exact value of the first solution by entering the command

```
rhs(solns[1]);
```

> The command `solns[1]` refers to the first element of the list `solns` which is `x=-`$\frac{\sqrt{5}-1}{2}$. This is an equation. The value needed is the right-hand side of this, which is extracted using the `rhs` command.

The operations introduced in the previous activity are summarised below.

Lists

Operation	Command	Example
Form a list	`[ , ,...]`	`A: [a,b,c];`
Extract an element of a list	`list[index]`	`A[2];`

Note: `...` indicates the list of arguments could continue.

As well as solving equations, you can also manipulate equations using Maxima. The next activity shows how you can do this.

Computer activity 22 Manipulating equations

(a) Assign the equation $3y + 4 = 3x^2 + 9x$ to the variable `eq`.

(b) Enter the following command.

```
eq-4;
```

This subtracts 4 from both sides of the equation `eq`.

(c) Now divide both sides of the rearranged equation by 3, by entering

```
%/3;
```

Remember that `%` represents the result of the last evaluated cell.

The variable `y` is now alone on the left-hand side of the equation.

We have rearranged the equation to make `y` the subject.

You can also use the `solve` command to change the subject of an equation: you solve the equation for the variable that you want to be the subject, as shown in the next activity.

Computer activity 23 Rearranging an equation

(a) Assign the equation $c = \dfrac{2a + b + 3c}{a - b}$ to the variable `eqn`.

(b) Rearrange the equation to make a the subject by entering

```
solve(eqn, a);
```

Here, the second argument of the `solve` command is the variable that you want to solve for; that is, the variable that you want to be the subject of the equation.

This use of the `solve` command is summarised below.

Changing the subject of an equation

Operation	Command	Example
Solve an equation for a variable	`solve(` *equation* `,` *variable* `)`	`solve(2*a*b-3*b=0, a);`

Another use of the `solve` command is to solve two or more simultaneous equations. To use the command in this way, you must enter the equations as a list, and enter the variables to be solved for as another list, as indicated below.

Solving simultaneous equations

Operation	Command	Example
Solve simultaneous equations	`solve(` *list of equations* `,` *list of variables* `)`	`solve([2*x+y=4, 3*x-2*y=-1], [x,y]);`

Computer activity 24 Solving simultaneous equations

Use Maxima to solve the simultaneous equations

$$3x^2 + 2y = 20$$
$$4x - 3y = -4$$

as described below.

(a) Assign the first equation to the variable `eq1` and the second equation to the variable `eq2`.

> Maxima might reorder the terms of the equations when displaying the output.

(b) Solve the pair of equations (`eq1` and `eq2`) for the two unknowns (x and y) by entering

```
solve( [eq1, eq2], [x,y] );
```

> Maxima's output is `[[x=2, y=4], [x=-26/9, y=-68/27]]`
>
> This tells you that there are two solutions to the equations: one solution is $x = 2$, $y = 4$ and the other is $x = -26/9$, $y = -68/27$. Notice that Maxima's output here is a list of lists.

3.3 Plotting graphs

You can plot graphs in wxMaxima by using the `wxplot2d` command. This command inserts your graph into your worksheet, as demonstrated in the activity below. Maxima plots the graph by calculating a large number of points on the curve and joining them up. The '`2d`' in the name of this command comes from the fact that its graphs are two-dimensional plots. The '`wx`' is because it is a wxMaxima command.

If you are using another interface to Maxima, you should use the command `plot2d` instead. You use it in the same way as `wxplot2d`, but it will plot the graph in a different computer window, which you then need to close before you can create another plot.

Computer activity 25 Plotting a graph

Plot the graph of the equation $y = 4x^2 - 8x + 2$ for values of x such that $-1 \leq x \leq 2$, as described below.

(a) Enter the command

```
wxplot2d(4*x^2-8*x+2, [x,-1,2]);
```

The first argument of the command is the expression to be plotted, and the second argument is a list that specifies the variable in the expression together with the least and greatest values that you want it to take in the graph.

wxMaxima displays the graph next to a line label beginning `%t`, such as `(%t1)`. This is how wxMaxima shows *intermediate* output, that is, output other than the final result of the command.

The `wxplot2d` command has no final result, just intermediate output, which is the graph.

(b) Change the vertical range of the graph to be $-5 \leq y \leq 5$ by including an additional argument `[y,-5,5]`.

That is, change the command to

```
wxplot2d(4*x^2-8*x+2, [x,-1,2], [y,-5,5]);
```

Notice the warning:

```
plot2d:  some values were clipped.
```

This means that some of the points on the graph with x-coordinates between -1 and 2 could not be displayed due to the range of values of y specified.

(c) Change the colour of the curve to green, by including the additional argument `[color, green]` after the other arguments of the `wxplot2d` command, but before the closing round bracket. Remember arguments are separated with commas, so include a comma before the new argument.

This argument is a list whose first element is a **keyword** indicating the property to be set, in this case `color` (with the American spelling!). The second element is the value of the property that you want to choose.

You can plot multiple graphs in the same diagram by including a list of expressions as the first argument of the command, as shown in the following activity.

Computer activity 26 Plotting multiple graphs

Suppose that the displacements s_1 and s_2 (in km) of two cars along a straight road from a given starting point are given by $s_1 = 15t^3 + 10t$ and $s_2 = 60t$, respectively, where t is the time (in hours) and $0 \leq t \leq 2$.

(a) Plot graphs of the displacement of the cars for $0 \leq t \leq 2$ by entering

```
wxplot2d([15*t^3+10*t, 60*t], [t,0,2]);
```

Notice that the first argument of the command `wxplot2d` is a list containing the two expressions to be plotted. In this case, the expressions involve the variable t, so the second argument specifies the range of values of t to be plotted.

The curves are automatically plotted in different colours, and a *legend* showing which expression corresponds to which curve is included at the top right-hand corner of the graph.

(b) Change the colours of the curves by adding the following additional argument to the command used above: `[color, green, magenta]`.

That is, change the command to

```
wxplot2d([15*t^3+10*t, 60*t], [t,0,2], [color, green, magenta]);
```

Maxima uses the first colour listed for the graph of the first expression listed and so on. If you do not include enough colours, Maxima starts using the listed colours again.

(c) Now change the labelling within the legend to be the variable names given to the expressions in the question, rather than the expressions themselves. Do this by including another argument

```
[legend, "s1", "s2"]
```

in a similar way to the colour argument; that is, within the round brackets of the `wxplot2d` command and separated from the other arguments by a comma.

In general, to change the legend text, you include an extra argument of `wxplot2d`: a list that contains the keyword `legend`, followed by the new pieces of text, each enclosed in double quote marks.

(d) Finally, change the labelling of the horizontal and vertical axes, by adding two more arguments to the command:

```
[xlabel, "time, t (h)"]
```

and

```
[ylabel, "displacement, s (km)"].
```

In general, to change the x-axis label, you include a further argument of `wxplot2d`: a list that contains the keyword `xlabel`, followed by the new label text enclosed in double quote marks. You can change the y-axis label in a similar way, using the keyword `ylabel`.

Changing the graph size

Finally, the size of the plot within your worksheet can be changed using the `wxplot_size` system variable. For example, entering

```
wxplot_size:[800,600];
```

changes the size of subsequently plotted graphs from the default setting of 600×400 pixels to 800×600 pixels.

Summary

The different ways in which you can use the `wxplot2d` command are summarised below, along with some of the optional arguments. If you are using a different interface to Maxima, you can use the `plot2d` command in the same ways.

Plotting graphs

Operation	Command	Example
Plot a graph	`wxplot2d(expression, horizontal range,...)`	`wxplot2d(x^2, [x,0,1]);` which plots the graph of x^2 for x between 0 and 1 `wxplot2d(x^2,[x,0,1],[y,0,2]);` which plots the graph of x^2 for x between 0 and 1, and with vertical axis values between 0 and 2
Plot several graphs	`wxplot2d(list of expressions, horizontal range,...)`	`wxplot2d([x^2, 2*x], [x,0,1]);` which plots graphs of x^2 and $2x$ for x between 0 and 1

Graph plotting options

Option	Argument	Example
Vertical range	`[y, , ]`	`[y, 0, 5]`
Curve colour	`[color, , ...]`	`[color, red]`
Legend text	`[legend, " ", ...]`	`[legend, "Car A"]` (a label should be listed for each curve plotted)
Turn legend off	`[legend, false]`	`[legend, false]`
Horizontal axis label	`[xlabel, " "]`	`[xlabel, "time, t (h)"]`
Vertical axis label	`[ylabel, " "]`	`[ylabel, "displacement, s (km)"]`
Change the size of the graph	`wxplot_size:[ , ]`	`wxplot_size:[800,600];`

Note: for a full list of the possible options, view the Maxima help obtained using the `?? plotting options;` command.

If you began studying Sections 2 and 3 of this Guide from Activity 31 of Unit 2, this completes your study of Unit 2.

4 Functions

You can define a function in Maxima using the `:=` operator, as shown in the following activity.

Computer activity 27 Defining functions

(a) Define the function f in Maxima, with the rule $f(x) = x^5 - 3x^2 + 4$, by entering the following command. (Use the characters `:` followed by `=` to type the `:=` operator.)

```
f(x):=x^5-3*x^2+4;
```

(b) Calculate the value of $f(2)$, by entering `f(2);`

(c) Now define the function g with rule

$$g(x) = \frac{x-1}{x-2}$$

and find the value of $g(3)$.

Remember to use brackets appropriately in the rule for g.

(d) Now try to use Maxima to find $g(2)$.

This gives an error.

This is because when Maxima tries to evaluate $g(2)$, it finds that this involves dividing by zero, so it returns an error message. The function g is undefined at $x = 2$.

(e) Enter `h(1);`

Maxima returns the expression `h(1)`, unevaluated.

Since the function `h` has not been defined, Maxima cannot calculate a value. It knows only that the value is `h(1)`.

(f) Enter `functions;`

This displays the value of the system variable `functions`, which is a list of all the functions that you have defined in this Maxima session.

Notice the difference between defining a *function*, which has an argument and which can be evaluated at different values of the argument, using `:=`, and assigning an expression to a *variable* (which has no argument) using `:`. As with variables, once you have defined a function in Maxima it is remembered by the system for the rest of your session, unless you remove it using the `kill` command that you met in Subsection 2.6 of this *Guide*.

Working with functions

Operation	Command	Example
Define a function	`:=`	`f(x):=2*x+3;`
Evaluate a function at a value	`function( )`	`f(1);`
List all user-defined functions	`functions`	`functions;`
Remove a defined function	`kill(function)`	`kill(f);`

You can plot the graph of a function using the `wxplot2d` command introduced in Subsection 3.3 of this *Guide*, as demonstrated in the next activity.

Computer activity 28 Plotting functions

(a) Plot the function `f` defined in Computer activity 27 over the range $-2 \leq x \leq 2$, by entering

```
wxplot2d(f(x), [x,-2,2]);
```

Note that the first argument must be `f(x)` rather than just `f`.

(b) Plot the function g defined in Computer activity 27 over the range $-5 \leq x \leq 5$.

Notice the vertical scale of the graph, and the large spike near $x = 2$.

The function g has the rule $g(x) = \dfrac{x-1}{x-2}$, so it is not defined at $x = 2$. Maxima has evaluated it at values of x close to 2, obtained large values and attempted to join up the corresponding points.

To obtain a better graph, you should restrict the range of the vertical axis. You are asked to do this next.

(c) Plot g again, for values of x in the range $-5 \leq x \leq 5$, but restricting the range of the vertical axis to between -5 and 5.

Remember, to restrict the vertical range to $-5 \leq y \leq 5$, include the argument `[y,-5,5]` in the `wxplot2d` command.

If you began studying this section from Activity 11 of Unit 3, return to the unit to continue your study of functions.

5 Exponential functions and logarithms

You saw in Subsection 2.3 of this *Guide* that the mathematical constant e is represented in Maxima by `%e`. The `%` symbol indicates that this is the name of a special quantity built into Maxima.

Computer activity 29 Working with exponentials and logarithms

(a) Find e^5 using Maxima, by entering

```
%e^5;
```

Since Maxima works symbolically, it does not return a decimal approximation.

(b) Find a decimal approximation to e^5 by entering

```
float(%e^5);
```

(c) An alternative way to find powers of e is to use the `exp` function.

Find e^5 by entering

```
exp(5);
```

Again, this is expressed exactly, as a power of e.

(d) The Maxima command for a natural logarithm is `log(  )`.

Use this to find a decimal approximation for $\ln 8$.

(e) Use the `solve` command introduced in Subsection 3.2 of this *Guide* to solve the equation

$$3^{2x} = 5.$$

You may be surprised to see two solutions to this equation.

One solution, $\dfrac{\log(5)}{2\log(3)}$, is the sort of solution that you would expect.

The other solution, $\dfrac{\log(-\sqrt{5})}{\log(3)}$, involves the logarithm of a negative number.

You have seen that only positive numbers have logarithms, but that applies when you are working only with real numbers. If you are working with complex numbers, then negative numbers do have logarithms.

The solution $\dfrac{\log(-\sqrt{5})}{\log(3)}$ is a complex number, as you can see more clearly by obtaining a decimal approximation for it. It contains the imaginary number i, which Maxima represents as `%i`. You will learn about complex numbers in Unit 12.

There is only one *real* solution to the equation.

The Maxima commands for the exponential function and logarithms are summarised below. Maxima has no command `ln( )`; remember that the command for a natural logarithm is `log( )`.

Exponentials and logarithms

Operation	Syntax	Example
Exponential, for example e^3	`%e^ ` *or* `exp( )`	`%e^3;` `exp(3);`
Natural logarithm, ln	`log( )`	`log(8);`

When you are working with logarithms in Maxima, the algebraic simplification commands that you met in Section 3 of this *Guide* can be useful. You can also use the commands shown below.

Simplifying expressions involving exponentials and logarithms

Operation	Command	Example
Simplify something involving exponentials and logarithms	`radcan( )`	`radcan(log(x^2));`
Combine logarithms	`logcontract( )`	`logcontract(log(a)+log(b));`

The name of the `radcan` command arises from the fact that it converts expressions involving *radicals*, that is, roots, powers and logarithms, into a canonical (or standard) form.

These commands are demonstrated in the following activity.

Computer activity 30 Simplifying expressions involving logarithms

Simplify the expressions

$$\frac{\ln(x^3+2x^2+x)-\ln(x)}{\ln(x+1)} \quad \text{and} \quad \ln(a)+2\ln(b)-\ln(c)$$

by following these steps.

(a) Define `u` to be the expression

$$\frac{\ln(x^3+2x^2+x)-\ln(x)}{\ln(x+1)}.$$

> Remember that the Maxima command for a natural logarithm is `log`.

(b) Simplify `u` using the following command.

```
radcan(u);
```

> Can you simplify this expression and obtain the same answer by hand?

(c) Define `v` to be the expression $\ln(a) + 2\ln(b) - \ln(c)$.

(d) Simplify `v` by using the following command.

```
logcontract(v);
```

If you began studying this section from Activity 48 of Unit 3, return to the unit to continue your study of functions.

6 Trigonometry

The Maxima commands for the trigonometric functions are given below. Like many computer systems, Maxima assumes that all angles are measured in radians. Notice that the names of the inverse trigonometric functions $\sin^{-1}$, $\cos^{-1}$ and $\tan^{-1}$ in Maxima are `asin`, `acos` and `atan`, respectively. This notation is used by many computer systems; it is short for arcsin, arccos and arctan, the alternative names often used for $\sin^{-1}$, $\cos^{-1}$ and $\tan^{-1}$.

Trigonometric functions

Function	Syntax	Example
sin	`sin( )`	`sin(1);`
cos	`cos( )`	`cos(3*%pi/2);`
tan	`tan( )`	`tan(%pi/4);`
cosec	`csc( )`	`csc(2);`
sec	`sec( )`	`sec(%pi/3);`
cot	`cot( )`	`cot(%pi/4);`
$\sin^{-1}$	`asin( )`	`asin(sqrt(3)/2);`
$\cos^{-1}$	`acos( )`	`acos(1/sqrt(2));`
$\tan^{-1}$	`atan( )`	`atan(1/2);`

Computer activity 31 Using trigonometric functions

Calculate the following using Maxima.

(a) $\cos\left(\frac{\pi}{6}\right)$

Remember that to enter the constant π in Maxima, you type `%pi`.

Maxima returns an exact answer, in the form of a surd. Remember that Maxima always uses exact values where possible, rather than decimal approximations.

(b) $\sin 45°$

Remember to convert the angle to radians, by multiplying 45° by $\frac{\pi}{180}$.

(c) $\sec(3.4)$

Here, since the angle is entered as a decimal number, Maxima returns a decimal approximation to the answer.

(d) $\sin^{-1}\left(\frac{\sqrt{3}}{2}\right)$

Maxima returns an exact answer, as a multiple of π.

(e) $\operatorname{cosec}\left(\frac{\pi}{7}\right)$

There is no simple exact answer here, so Maxima leaves this expression unevaluated.

To find a decimal approximation, you can use the command

```
float( csc(%pi/7) );
```

(f) $\tan\left(-\frac{\pi}{12}\right)$

Again, Maxima leaves this expression unevaluated, but it simplifies it slightly, using the fact that $\tan(-\theta) = -\tan\theta$.

(g) $\cos^{-1}(2.0)$

Here, since the argument of the function is a decimal number, Maxima returns a decimal approximation to the answer.

You might be surprised that it returns an answer at all, since 2.0 is not in the domain of $\cos^{-1}$.

Notice that the answer is a multiple of `%i`. In Maxima, `%i` represents the imaginary number i whose square is -1, which you will learn about in Unit 12. Although there is no *real* number y such that $\cos y = 2$, there is an imaginary number with this property, and Maxima returns this value.

If you are working only with real numbers, you should ignore any Maxima answers that contain the constant `%i`.

(h) $\cos^{-1}(2)$

This expression is equivalent to the expression in part (g), but does not contain a decimal number, so Maxima tries to find an exact answer. Since there is no simple exact answer, Maxima leaves the expression unevaluated.

This hides the fact that the answer is not a real number. You need to take care when using inverse trigonometric functions.

You can solve equations involving trigonometric functions in Maxima, using the `solve` command that you met in Subsection 3.2 of this *Guide*. However, Maxima will return only one solution to such an equation, even though many may exist. It warns you of this, as shown in the following activity.

Computer activity 32 Solving trigonometric equations

(a) Solve the equation $\cos(x) - 0.6 = 0$ by entering

```
solve(cos(x)-0.6=0);
```

Since the `solve` command always tries to find exact solutions, Maxima first converts the decimal number -0.6 to the exact fraction $-\frac{3}{5}$. It tells you that it has done this.

It also displays the following warning to tell you that there might be other solutions.

```
solve:  using arc-trig functions to get a solution.
Some solutions will be lost.
```

Then it returns a single solution.

(b) Find a decimal approximation to the solution obtained in part (a), by entering

```
float(%)
```

Remember, `%` represents the answer of the last evaluated calculation.

(c) Solve the equation $\sin(3x + 4) = \frac{1}{2}$.

(d) Solve the equation $\cos(2x + 4) - \sin(x/3) = 0$.

Maxima returns the original equation, slightly rearranged but unsolved. This is because the `solve` command could not find an exact solution to this equation.

In circumstances such as Computer activity 32(d) when the `solve` command fails to find the exact solution of an equation, you can use another Maxima command, `find_root`, to try to find an approximate solution.

Whereas the `solve` command manipulates the equation mathematically to find a solution, the `find_root` command performs a series of numerical calculations to find a decimal approximation to a solution. This is known as solving the equation **numerically**, or finding a **numerical approximation** to a solution.

You know from Unit 2 that the solutions of an equation of the form $f(x) = 0$ are the x-coordinates of the points at which the graph of $y = f(x)$ crosses the x-axis, as illustrated in Figure 10.

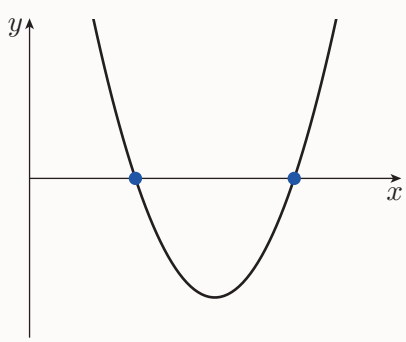


Figure 10 The graph of $y = f(x)$ with the solutions of $f(x) = 0$ marked

The `find_root` command uses the following property of functions. If f is a function and $[a, b]$ is an interval included in its domain such that one of $f(a)$ and $f(b)$ is negative and the other is positive, then there is at least one solution of the equation $f(x) = 0$ in the interval (a, b). Figure 11 illustrates this property for two different functions. This property holds as long as the graph of f is *continuous* on the interval $[a, b]$; informally, this means that you can draw the graph of f for input values in the interval $[a, b]$ without taking your pen off the paper.

The `find_root` command looks for a single solution of an equation of the form $f(x) = 0$ that lies within an interval that you specify. The command requires that the values of f at each endpoint of the interval have different signs. If this is not the case, then Maxima gives an error message.

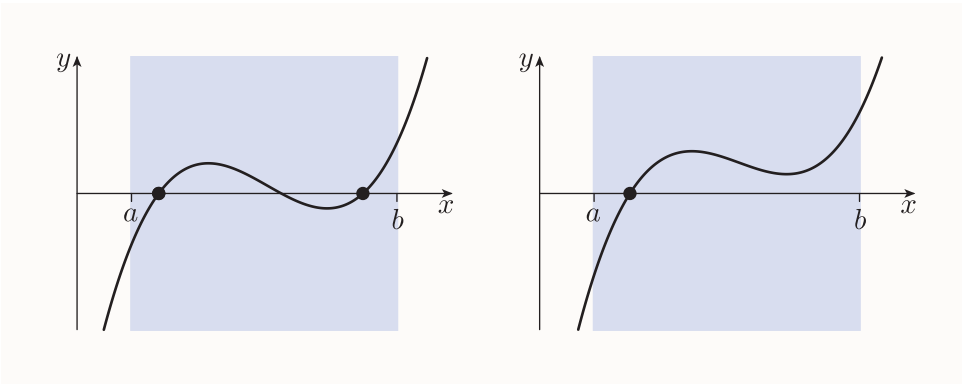


Figure 11 There is at least one solution of $f(x) = 0$ in the interval (a, b) if $f(a)$ and $f(b)$ have different signs

Solving equations numerically

Operation	Command	Example
Solve an equation numerically	`find_root(` `expression` `,` `variable` `,` `interval start value` `,` `interval end value` `)`	`find_root(2*x^3-1,x,0,1);`

Note 1: the command finds a value of the variable at which the expression is zero.

Note 2: the values that the expression takes at the endpoints of the interval must have different signs.

Computer activity 33 Solving equations numerically

(a) Find a numerical approximation to a solution of the equation

$$\cos(2x + 4) - \sin\left(\frac{x}{3}\right) = 0,$$

as follows.

(i) First, define the function f in Maxima, with the rule

$$f(x) = \cos(2x + 4) - \sin\left(\frac{x}{3}\right).$$

Remember, to define a function use `:=`.

(ii) Next, plot the graph of $f(x)$ over the interval $-2 \leq x \leq 2$, say. This will help you to find an interval that contains a solution.

Remember, you can plot graphs using the `wxplot2d` command.

From the graph, you can see that $f(0) < 0$ and $f(1) > 0$, so there is a solution in the interval $(0, 1)$.

If there were no solution in the interval over which you plotted the function, then you could plot the function over a larger interval, and so on, until you find a suitable interval, if one exists.

(iii) Find a numerical approximation to the solution of the equation in the interval $(0, 1)$ by entering

```
find_root(f(x), x, 0, 1);
```

Remember to type the underscore symbol (_) in the middle of the command name.

Notice that the first argument of the `find_root` command is an *expression* (or a function defined by an expression) not an *equation*. Maxima finds an input value at which the expression takes the value 0.

(b) Using similar methods, find a solution of the equation $\sin^2(x + 2) = 3x$.

Hint: consider the function $g(x) = \sin^2(x + 2) - 3x$ and find a value of x for which $g(x) = 0$.

Remember that $\sin^2(x + 2)$ means $(\sin(x + 2))^2$. You can enter this expression in Maxima by typing either `(sin(x+2))^2` or `sin(x+2)^2`.

Maxima can use trigonometric identities to rewrite expressions involving trigonometric functions in different forms.

These behave like the algebraic simplification commands you met in Section 3 of this *Guide*. In particular, you need to guide Maxima in how to conduct such rearrangements. Commands for doing this are demonstrated in Computer activities 34 and 35.

Computer activity 34 Manipulating trigonometric expressions

(a) Enter

```
trigreduce( sin(x)^2 );
```

The `trigreduce` command attempts to express powers of sines and cosines of x in terms of sines and cosines of multiples of x. It *reduces* the powers of the functions.

(b) Enter

```
trigexpand(%);
```

> Remember that `%` represents the result of the previous calculation.
>
> The `trigexpand` command attempts to express trigonometric functions of sums, differences and multiples of angles in terms of trigonometric functions of the individual angles.

(c) Enter

```
trigsimp(%);
```

> The `trigsimp` command attempts to simplify expressions involving trigonometric functions.
>
> This returns you to the expression that you started with!

Computer activity 35 Simplifying rational trigonometric expressions

Simplify $\dfrac{\sin(4x)}{\sin(x)}$ by entering

```
trigrat( sin(4*x)/sin(x) );
```

> The `trigrat` command attempts to simplify algebraic fractions involving trigonometric functions.

Trigonometric manipulation commands

Operation	Command	Example
Expand trigonometric functions of sums, differences and multiples of angles	`trigexpand( )`	`trigexpand(sin(A+B));`
Express powers of sines and cosines in terms of sines and cosines of multiple angles	`trigreduce( )`	`trigreduce(sin(x)^2);`
Simplify trigonometric expressions	`trigsimp( )`	`trigsimp(sin(x)^2+cos(x)^2);`
Simplify algebraic fractions containing trigonometric functions	`trigrat( )`	`trigrat((sin(2x))/cos(x));`

You can use these commands, and others that you learned previously, in the final activity of this section.

Computer activity 36 Finding a new trigonometric identity

(a) Find an expression for $\cos(5x)$ in terms of products of $\sin(x)$ and $\cos(x)$.

(b) Plot the graphs of $\cos(5x)$ and the expression you found in (a), to check that they look the same.

If you began studying this section from Activity 37 of Unit 4, this completes your study of Unit 4.

7 Plotting circles

In this section you will learn how to use Maxima to plot circles.

In Subsection 3.3 of this *Guide*, you learned how to use the `wxplot2d` command to plot curves. You can use this command to plot a curve only if the equation of the curve is in *explicit* form; that is, only if it is written in the form $y = f(x)$, where f is a function.

To plot a curve represented by an equation in *implicit form*, such as $x^2 + y^2 = 1$, you need to use the `wximplicit_plot` command. This is demonstrated in Computer activity 37.

The `wximplicit_plot` command is not part of the basic Maxima collection of commands, so you have to add it by loading an additional Maxima **package**, the `implicit_plot` package, which was installed on your computer when you installed Maxima. The Maxima command to load an additional package is `load`, which is also demonstrated in Computer activity 37. Each package needs to be loaded only once per Maxima session.

If you are using a Maxima interface other than wxMaxima, then you should use the `implicit_plot` command instead of the `wximplicit_plot` command. You load and use it in exactly the same way as `wximplicit_plot`.

Computer activity 37 Plotting a circle

Plot the circle represented by the equation $x^2 + y^2 = 1$, as follows.

(a) Load the `implicit_plot` package by entering

```
load(implicit_plot);
```

Remember to type the underscore symbol (_) in the middle of the package name.

This package provides the `wximplicit_plot` command.

(b) Plot the circle by entering

```
wximplicit_plot(x^2+y^2=1, [x,-2,2], [y,-2,2]);
```

Again, remember to type the underscore symbol (_) in the middle of the command name.

The first argument of the command is the equation to be plotted. Notice that in Maxima, equations are given using the usual equals sign.

The second argument specifies the variable to be plotted on the horizontal axis, and the range of values of that variable to be plotted.

The third argument specifies the variable (and its range) to be plotted on the vertical axis. You must include this third argument.

The circle with equation $x^2 + y^2 = 1$ has centre $(0, 0)$ and radius 1. So the circle lies within $-2 \leq x \leq 2$ and $-2 \leq y \leq 2$ and hence these seem suitable ranges to use.

The curve that Maxima plots, however, does not look very circular! This is discussed below.

Note that the first argument of the `wximplicit_plot` command is an *equation*, whereas the first argument of the `wxplot2d` command is an *expression* (or a function or variable representing an expression). If an expression is given as the first argument, `wximplicit_plot` will plot the graph of the equation obtained by setting the expression equal to zero.

The curve that you plotted in Computer activity 37 does not look circular because it has different scales on the axes. Whatever ranges of x- and y-values you specify for a plot, by default Maxima makes the y-axis appear $\frac{3}{5}$ as long as the x-axis.

To specify the use of equal scales, you can add the following additional argument to the `wximplicit_plot` command:

```
same_xy
```

Note that in some older releases of Maxima, the option needed to specify equal scalings on coordinate axes was

```
[gnuplot_preamble, "set size ratio -1"]
```

If the `same_xy` option does not change the scales of your graph, then use this option instead.

Computer activity 38 Plotting a circular circle!

Plot the circle $x^2 + y^2 = 1$, using equal scales on the x- and y-axes by entering

```
wximplicit_plot(x^2+y^2=1, [x,-2,2], [y,-2,2], same_xy);
```

Including the `same_xy` (or `gnuplot_preamble`) argument in the `wximplicit_plot` command can be tiresome if you are plotting lots of curves and need equally scaled x- and y-axes for each. An alternative is to change the default behaviour of Maxima so that it uses axes with equal scales for the rest of your session. You will see how to do this in the following activity.

Computer activity 39 Plotting another circular circle!

Plot the circle $(x-1)^2 + y^2 = 3$, using equal scales on the x- and y-axes as follows.

(a) Tell Maxima to always use equal scales on the x- and y-axes by entering

```
set_plot_option(same_xy)$
```

This sets `same_xy` to be a default option for all plots during the rest of your Maxima session. The line ends with `$` to prevent the display of the output of the command. Without this, the output consists of a long list of all the current default settings, which can safely be ignored.

If you are using an older release of Maxima that uses the `gnuplot_preamble` option, use

```
set_plot_option([gnuplot_preamble, "set size ratio -1"])$
```

instead.

(b) Plot the circle by entering

```
wximplicit_plot((x-1)^2+y^2=3, [x,-1,3], [y,-2,2]);
```

If you wish to stop plotting graphs with equal scales and restore the default Maxima plotting behaviour, then set the `same_xy` plotting option to be `false` using

```
set_plot_option([same_xy, false])$
```

If you are using an older version of Maxima, the command needed to reset the default view is

```
set_plot_option([gnuplot_preamble, ""])$
```

To plot more than one curve at once you can list their equations within square brackets as the first argument of `wximplicit_plot`, as you did with the `wxplot2d` command. The equations do not necessarily have to be in implicit form. This is demonstrated in the following activity.

Computer activity 40 Plotting a line and a circle

Plot the line $y = 3x - 5$ and the circle $(x-2)^2 + (y-3)^2 = 2$ as follows.

(a) For simplicity, assign the equation of the line to the variable `line`, and the equation of the circle to the variable `circle`.

> Remember, to assign an expression to a variable, use the colon (`:`) operator. So to assign the equation of the line to the variable `line`, use
>
> ```
> line:y=3*x-5;
> ```

(b) Plot the line and circle on axes with equal scales by entering

```
wximplicit_plot([line, circle], [x,0,5], [y,0,5]);
```

> The first argument is a list of the equations to be plotted. Since you have assigned the equations to variables, you can simply list the names of the variables.
>
> Since the circle has centre $(2, 3)$ and radius $\sqrt{2} \approx 1.41$, the ranges $0 \leq x \leq 5$ and $0 \leq y \leq 5$ are large enough for the whole circle to be plotted.

The commands introduced in this section are summarised below.

Loading additional packages

Operation	Command	Example
Load a package	`load(package name)`	`load(implicit_plot);`

Plotting curves represented by equations in implicit form

Operation	Command	Example
Plot a curve represented by an equation in implicit form	`wximplicit_plot(equation, horizontal range, vertical range, ...)`	`wximplicit_plot(x^2+y^2=1, [x,-1,1], [y,-1,1]);`
Plot several curves represented by equations in implicit form	`wximplicit_plot( list of equations, horizontal range, vertical range, ...)`	`wximplicit_plot( [x^2+2*y^2=4, y=x^3], [x,-1,1], [y,-1,1]);`

Note 1: the `implicit_plot` package needs to be loaded before this command can be used.

Note 2: the first argument is an equation or a list of equations. Alternatively it can be an expression, or a list of expressions, in which case the equation formed by setting the expression equal to zero is plotted.

Note 3: equations in explicit form can also be plotted using this command.

Graph plotting options

Option	Argument	Example
Set axes to have equal scales *(in older versions of Maxima)*	`same_xy [gnuplot_preamble, "set size ratio -1"]`	`same_xy [gnuplot_preamble, "set size ratio -1"]`

Setting default plotting options

Operation	Command	Example
Set the default plotting options	`set_plot_option(option)`	`set_plot_option(same_xy)`

Circles are not the only type of curve that can be represented by equations in implicit form. In the next activity you will plot some other equations of this type and see the curves that they represent.

Computer activity 41 Plotting equations given in implicit form

Plot the curves represented by the following equations, using equal axis scales for each curve. Make sure you use suitable ranges of x and y.

> Remember, you can use the editing facilities of wxMaxima to avoid having to type the whole `wximplicit_plot` command each time.

(a) $x^2 + 4y^2 = 1$ (b) $x^2 - 4y^2 = 1$ (c) $y^2 = x^3 - 2x + 1$

(d) $y^2 = x^3 - 2x + 5$ (e) $y^6 = x^2 - x^6$ (f) $(x^2 + y^2 - 1)^3 = x^2y^3$

If you began studying this section from Activity 16 of Unit 5, return to the unit to continue your study.

8 Differentiation

The Maxima command for differentiating an expression is `diff`, which is demonstrated in the following activity. This command finds derivatives by using the derivatives of standard functions together with the sum rule, constant multiple rule, product rule, quotient rule and chain rule.

Sometimes Maxima may give the result in a different, but equivalent, form to the expression that you might obtain when differentiating by hand. If this happens, then you might like to check that the two answers are equivalent. You can do this by finding the difference between the two answers and checking that it simplifies to zero. A rougher check is to use Maxima to plot graphs of the two answers and check they appear to be the same.

Computer activity 42 Demonstrating differentiation with Maxima

(a) Find the derivative of $f(x) = x^4 + 5x^2 + 15$, by entering

```
diff(x^4+5*x^2+15, x);
```

The first argument of the `diff` command is the expression to be differentiated, and the second argument is the variable to differentiate with respect to.

You must include the second argument. If you do not, then the answer returned by Maxima will be the derivative multiplied by `del(x)`. This notation is outside the scope of MST124.

As you would expect, the first argument can also be a variable that you have previously defined to be an expression, or a function you have previously defined.

(b) Define the variable `p` to be equal to $\dfrac{t^2}{\sin(4t)}$, and then find its derivative with respect to t by entering

```
diff(p,t);
```

(Remember, to assign a value to a variable you use a colon (`:`).)

Using `diff` in this way can help you check the expression to be differentiated has been entered correctly before you differentiate it.

(c) Define the function $g(u) = (u^2 + 3)\ln(u^2)$, and find its derivative using

```
diff(g(u),u);
```

(Remember, to define a function you use `:=`, and the Maxima command for natural logarithms is `log`.)

Notice that the first argument given here is `g(u)`, not just `g`.

(d) Find the second derivative of the expression `p` defined above, by using

```
diff(p,t,2);
```

The third argument of `diff` specifies the number of times that the expression is to be differentiated.

(e) Find the third derivative of $g(u)$ defined above.

The `diff` command is summarised below. You can practise using it in the next activity.

Differentiation

Operation	Command	Example
Differentiate	`diff( , variable )`	`diff(sin(x^2),x);`
Differentiate multiple times	`diff( , variable , positive integer )`	`diff(log(x),x,3);`

Computer activity 43 Differentiating with Maxima

Use Maxima to differentiate the following, simplifying the answer where possible.

Remember, you can enter e^x in Maxima either as `%e^x` or `exp(x)`.

(a) $2x^6 + \dfrac{5}{x}$ (b) $\ln(e^x + x)$ (c) $\dfrac{\tan(t) - t^2}{e^t}$

You can use the `diff` command to assign the derivative of a function to another named function. This is useful when you want to evaluate the derivative at a particular input value, for example.

The next activity shows you how to do this – unfortunately it's not completely straightforward. In the activity, the original function is denoted by `f` and its derivative is denoted by `df`, rather than by `f'`. This is because you cannot use `f'` as a function name in Maxima.

Computer activity 44 Assigning a derivative to a function

(a) Define the function $f(x) = \dfrac{\sin(x)}{e^x + 1}$.

(b) Try to define `df(x)` to be the derivative of $f(x)$, by entering

```
df(x):=diff(f(x),x);
```

(c) Try to find the value of the derivative of f at $x = 3$ by evaluating `df(x)` at $x = 3$; that is, try entering

```
df(3);
```

This gives an error.

The cause of this error lies in the previous command:

```
df(x):=diff(f(x),x);
```

The problem is that when you enter this command, Maxima does not perform the differentiation. Instead, it just notes that `df(x)` means `diff(f(x),x)`. Then, when you enter `df(3)`, Maxima understands this to mean `diff(f(3),3)`, which is meaningless, as you cannot differentiate with respect to a particular number.

To force Maxima to perform the differentiation, you can use the syntax `''(  )` (which contains two single quotation marks, not a double quotation mark, and a pair of round brackets). This is demonstrated below. You need to use this syntax whenever you want to assign the result of a Maxima command to a function.

(d) Define `df(x)` to be the *result* of differentiating f with respect to x, by entering

```
df(x):=''(diff(f(x),x));
```

Remember that `''` is two *separate* quotation marks. Use the upright quotation mark on your keyboard (`'`), not the sloping one (`` ` ``). Don't forget to include the outer round brackets too.

Notice that this time the output line shows that `df(x)` is defined to be the result of differentiating `f(x)`.

(e) Find the value of the derivative of f at $x = 3$ by entering

```
df(3);
```

If you want a decimal approximation to the result, use the `float` command.

(f) Use a similar approach to find the derivative of $\sin(x)e^{\cos(x)}$ at $x = 1$, giving your answer as a decimal number to 3 significant figures.

The command used to force the evaluation of the derivative in Computer activity 44 is summarised below.

Forcing the evaluation of a command

Operation	Command	Example
Force the evaluation of a command	`''(  )`	`g(x):=''(diff(x^4,x));`

An alternative approach to Computer activity 44 parts (d) and (e) is to assign the derivative of f to a variable rather than use the derivative to define a function. You can do this by entering, for example

```
df:diff(f(x),x);
```

This calculates an expression for the derivative of f at x and assigns it to the variable `df` (the `''(  )` command is not required).

You can then evaluate the derivative at 3 by substituting the value 3 for x, which you can do by entering

```
subst(3,x,df);
```

If you began studying this section from Activity 22 of Unit 7, return to the unit to continue your study.

9 Integration

The Maxima command to integrate an expression is `integrate`. This is demonstrated in the following activity.

As with differentiation, the results that you obtain from integrating using Maxima might be in a different algebraic form to the expression that you might obtain when integrating by hand. To show that the two answers are equivalent integrals, you could find the difference between the two answers and check that it simplifies to a constant. A rougher check is to use Maxima to plot graphs of the two answers (taking the arbitrary constant to be zero, for example) and check they seem to be vertical translations of each other.

Computer activity 45 Demonstrating integration with Maxima

(a) Evaluate $\int 2x^2 + e^{3x}\,\mathrm{d}x$ by entering

```
integrate(2*x^2+exp(3*x), x);
```

The first argument of `integrate` is the expression to integrate, and the second is the variable to integrate with respect to.

The output returned by Maxima does not include an arbitrary constant. That is, Maxima gives an antiderivative of the expression, not its indefinite integral. When you write down an indefinite integral that you have found using Maxima, you need to remember to add an arbitrary constant yourself.

(b) Evaluate the definite integral $\int_2^3 \frac{1}{x}\,\mathrm{d}x$ by entering

```
integrate(1/x, x, 2, 3);
```

The third and fourth arguments of `integrate` are the lower and upper limits of integration, respectively.

Notice that, as usual, Maxima returns an exact result. Use the `float` command to find a decimal approximation, if required.

(c) Define the function $p(x) = \dfrac{x}{x^2+1}$, and define `q(x)` to be an antiderivative of p.

If you try to define `q(x)` using

```
q(x):=integrate(p(x),x);
```

then Maxima does not actually do the integration, but just notes that `q(x)` means $\int p(x)\,\mathrm{d}x$.

As with differentiation, to force Maxima to perform the integration you need to use the `''( )` command and enter

```
q(x):=''(integrate(p(x),x));
```

Integration

Operation	Command	Example
Integrate (find an antiderivative)	`integrate( , variable )`	`integrate(x^2,x);`
Evaluate a definite integral	`integrate( , variable , lower limit , upper limit )`	`integrate(sin(x),x,0,1);`

You can practise using the `integrate` command in the following activity.

Computer activity 46 Using integration

Use Maxima to find the following.

(a) $\int (x-3)(x-1)\,\mathrm{d}x$ (b) $\int \frac{1}{\sqrt{9-t^2}}\,\mathrm{d}t$ (c) $\int_0^{\pi} e^t \sin t\,\mathrm{d}t$

It is not always possible to express the result of integrating a function in terms of the mathematical functions you have previously met. The following activity illustrates some of the things that you may encounter when integrating functions using Maxima.

Computer activity 47 Calculating harder integrals

Try to calculate the following using Maxima.

(a) $\int e^{\sin(x)}\,\mathrm{d}x$ (b) $\int e^{-x^2}\,\mathrm{d}x$ (c) $\int \frac{e^x}{x}\,\mathrm{d}x$

Maxima was unable to find an antiderivative for part (a), and so returned the integral uncalculated. In parts (b) and (c) answers were returned containing the function `erf`, which is known as the *Error function*, and the function `gamma_incomplete`, the *incomplete gamma function*. These are examples of what are known as *special functions* in mathematics. They cannot be simply expressed in terms of functions you have already met, but many can be defined in terms of antiderivatives of simpler functions.

Sometimes you may want to integrate a function that contains a variable other than the one you are integrating with respect to. For example, the integral $\int x^n\,\mathrm{d}x$ contains the variable n as well as the variable x. When you use Maxima to find such an integral, you sometimes need to provide information about the other variable, as demonstrated in the following activity.

Computer activity 48 Giving Maxima more information

(a) Try to use Maxima to find the integral $\int x^n\,\mathrm{d}x$, as follows.

(i) First, ensure that `n` has no predefined value, by entering

```
kill(n);
```

(ii) Then, enter

```
integrate(x^n, x);
```

Maxima asks the question: `Is n equal to -1?`

This is because the answer to $\int x^n \, dx$ depends on the value of n.

If $n = -1$ (so $n + 1 = 0$), the integral is $\int \frac{1}{x} \, dx$, which is equal to $\ln|x| + c$, whereas if n is any other value, then the integral is equal to $\frac{x^{n+1}}{n+1} + c$. So Maxima cannot calculate the integral without knowing more about n.

Enter the answer `n;`
(which is short for `no`)

You could also answer the question with `y;` to mean `yes`.

Maxima displays the expected result.

(b) An alternative approach is to give Maxima information about n *before* you use the `integrate` command. You can do this as follows.

(i) Tell Maxima that $n \neq -1$ by entering

```
assume(notequal(n,-1));
```

The `assume` command tells Maxima information about variables. For example, `assume(a>0);` tells Maxima that `a` is positive.

Maxima has no simple symbol for 'not equal to', so you have to use the `notequal` command for this, as above.

(ii) Now integrate x^n using

```
integrate(x^n, x);
```

No question is asked this time, since Maxima knows all it needs to know about `n`.

(iii) Enter

```
facts();
```

to see all the additional facts about variables known to Maxima.

This lists all such facts. The only one shown should be the one you gave earlier.

(iv) Tell Maxima to forget about the assumption on `n` by entering

```
forget(notequal(n,-1));
```

(v) Type `facts();` again to see all the facts now known.

No facts are displayed, since Maxima has now forgotten the assumption about n.

The commands introduced in the previous activity are summarised below, together with the `equal` command, which is used in a similar fashion to `notequal`.

Facts about variables

Operation	Command	Example
Make an assumption about a variable	`assume( )`	`assume(a>0);`
State two things are equal	`equal( , )`	`assume(equal(n,3));`
State two things are not equal	`notequal( , )`	`assume(notequal(n,-1));`
Forget a property	`forget(property)`	`forget(a>0);`
List all known facts	`facts()`	`facts();`
List all known facts about a particular variable	`facts(variable)`	`facts(a);`

Computer activity 49 Using facts

Calculate the integral

$$\int \frac{1}{x^2 + a}\,\mathrm{d}x, \quad \text{where } a > 0.$$

You saw in Computer activity 47 that Maxima cannot always find an antiderivative for a function. Even if Maxima cannot find an antiderivative of a function f, it can still find an *approximate* value for a definite integral of the form $\int_a^b f(x)\,\mathrm{d}x$.

In Unit 8, you saw how the value of the definite integral can be approximated by breaking the area between the graph of the function and the x-axis into a number of rectangles. Maxima uses a more sophisticated version of this basic method to calculate approximate values of definite integrals.

You will learn the Maxima command for this in the following activity.

Computer activity 50 Approximate values of definite integrals

(a) Try to calculate $\int_0^1 \ln(\cos(x^2))\,\mathrm{d}x$ by entering

```
integrate(log(cos(x^2)), x, 0, 1);
```

The integral is returned unevaluated, since Maxima cannot calculate it.

(b) Find an approximate value of the integral by entering

```
quad_qags(log(cos(x^2)), x, 0, 1);
```

The `quad_qags` command calculates an approximate value for a definite integral.

The 'quad' part of the name of this command arises from the fact the process of finding the value of a definite integral (or any area) is sometimes called **quadrature**. The letters `q`, `a`, `g` and `s` in the second half of the name specify the particular method Maxima uses to find the approximate value of the definite integral. This method gives good results for a wide variety of integrals.

Notice that the output of the `quad_qags` command is a list. The first element of the list is the approximate value of the definite integral, the second element is an estimate of the accuracy of the approximation, the third element is the number of values at which the function to be integrated was evaluated during the calculation, and the final element is an error code. An error code of 0 means that no errors occurred.

(c) Find an approximate value for the definite integral $\int_0^1 e^{-t^2}\,\mathrm{d}t$.

Finding approximate values for definite integrals

Operation	Command	Example
Find an approximate value of a definite integral	`quad_qags(` *expression* `,` *variable* `,` *lower limit* `,` *upper limit* `)`	`quad_qags(exp(-x^2),x,0,1);`

If you began studying this section from Activity 47 of Unit 8, return to the unit to continue your study.

10 Matrices

In this section you will learn how to use Maxima for matrices.

10.1 Matrix algebra

There are several ways to input matrices in Maxima. Two of these are demonstrated in the following activity.

Computer activity 51 Entering matrices

(a) Assign the matrix $\begin{pmatrix} 2 & -3 & 3 \\ -2 & 1 & 2 \\ 1 & -1 & 4 \end{pmatrix}$ to the variable `P` by entering

```
P:matrix( [2,-3,3], [-2,1,2], [1,-1,4] );
```

The `matrix` command allows a matrix to be defined. You give each row of the matrix as a list (that is, within square brackets with the elements separated by commas).

By default, wxMaxima displays matrices using square brackets:

$$\begin{bmatrix} 2 & -3 & 3 \\ -2 & 1 & 2 \\ 1 & -1 & 4 \end{bmatrix}$$

If you are using a different interface to Maxima, or have chosen different display settings, matrices may be displayed as follows.

```
[  2  -3  3 ]
[           ]
[ -2   1  2 ]
[           ]
[  1  -1  4 ]
```

(b) If you are using wxMaxima, you can also input a matrix using an on-screen form. Use this form to assign the matrix

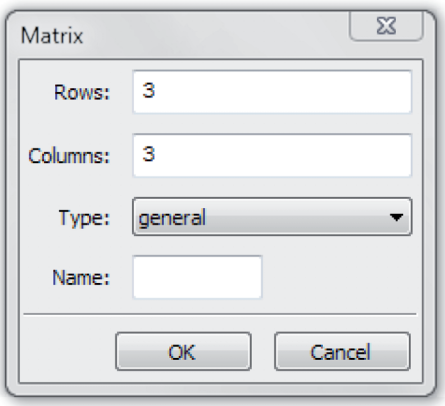
$$\begin{pmatrix} 5 & -3 \\ -1 & -2 \\ 0 & 4 \end{pmatrix}$$

to the variable `Q` as follows. (If you are not using wxMaxima, then enter this matrix using the `matrix` command, as in part (a).)

(i) From the `Algebra` menu select `Enter Matrix...`.

Make sure you select `Enter Matrix...`, not `Generate Matrix...`.

A window opens requesting properties of the matrix, as shown below.

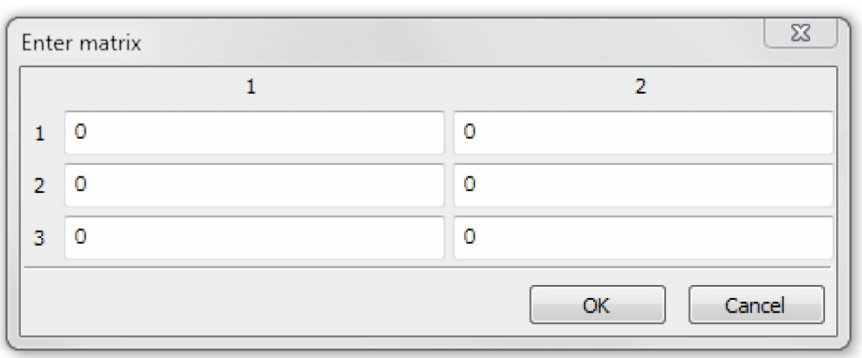


(ii) The matrix for this example has size 3×2, so enter 3 as the number of rows and 2 as the number of columns, replacing the default values shown.

It is also possible to give information on the type of matrix, but for now leave this set to `general`.

Finally, enter `Q` as the variable name to which the matrix will be assigned, and click 'OK'.

This opens a second window requesting the elements of the matrix.



(iii) Complete entry of the matrix by filling in the elements in this second window. You may like to use the `Tab` key (⇥) to move the cursor between the different elements. When you have finished, click 'OK'.

Completion of this on-screen form has the effect of including a `matrix` input line in your worksheet, followed by the appropriate output.

(c) Assign the matrix $\begin{pmatrix} 1 & -2 & -1 \\ 4 & 7 & 0 \\ -5 & 1 & 3 \end{pmatrix}$ to the variable `R` using an appropriate method.

The typed command for entering a matrix is summarised below.

Entering matrices

Operation	Command	Example
Specify a matrix	`matrix(row,row,...)`	`matrix([1,2],[3,4]);`

Note: matrices can also be entered using the `Enter Matrix...` option of the `Algebra` menu.

Once a matrix has been input, you can obtain its size and individual elements by using the commands demonstrated in the following activity.

Computer activity 52 Using matrices

(a) Find the size of the matrix `Q` defined in Computer activity 51 by entering

```
matrix_size(Q);
```

The output is a list containing the number of rows and the number of columns of the matrix.

(b) Extract the element in the third row and second column of `R` by entering

```
R[3,2];
```

Alternatively, `R[3][2]` gives the same result.

Notice the use of square brackets to reference elements of a matrix is the same syntax as used in Section 3.2 of this *Guide* to obtain individual elements of a list.

(c) Extract the third row of `R` by entering

```
R[3];
```

(d) Change the element in the third row and second column of `R` to be 0 by entering

```
R[3,2]:0;
```

then display the revised matrix by entering

```
R;
```

You can use commands of this type to correct any errors that you might make when entering a matrix.

Using matrices

Operation	Command	Example
Size of a matrix	`matrix_size(matrix)`	`matrix_size(A);`
Row of a matrix	`matrix[row]`	`A[2];`
Element of a matrix	`matrix[row,column]`	`A[3,4];`
	or `matrix[row][column]`	`A[3][4];`

The syntax to use for matrix operations in Maxima is given below. You can practise using them in Computer activity 53.

Matrix operations

Operation	Syntax	Example
Matrix addition	`+`	`A+B;` where `A` and `B` are matrices of the same size
Matrix subtraction	`-`	`A-B;` where `A` and `B` are matrices of the same size
Scalar multiplication	`*`	`2*A;`
Matrix multiplication	`.`	`A.B;` where `A` and `B` are matrices of appropriate sizes
Matrix powers	`^^`	`A^^3;` where `A` is a square matrix

Warning

Remember to use `.` (not `*`) for matrix multiplication and `^^` (not `^`) to find the power of a matrix.

If you use the symbol `*` when multiplying two matrices or the symbol `^` to find the power of a matrix, then Maxima may return an answer, but in general it will be the *wrong* answer. (It will be an answer found using a different matrix operation.)

Computer activity 53 Matrix operations

(a) Define the following matrices in Maxima

$$\mathbf{P} = \begin{pmatrix} 2 & 0 \\ 3 & -1 \\ -4 & 6 \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} 0 & 3 \\ 5 & -4 \\ 1 & 3 \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix},$$

$$\mathbf{S} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} 3 & 4 \end{pmatrix}.$$

(b) Find $\mathbf{P} + \mathbf{Q}$.

The matrices $\mathbf{P}$ and $\mathbf{Q}$ are the same size, so this calculation is possible.

(c) Find $\mathbf{P} - \mathbf{R}$.

The matrices $\mathbf{P}$ and $\mathbf{R}$ are not the same size, so this calculation is not possible. Maxima gives an error message:

```
fullmap:  arguments must have same formal structure.
 -- an error.  To debug this try:  debugmode(true);
```

(d) Find $2\mathbf{P}$.

This is a scalar multiple of a matrix, so use the `*` operator.

(e) Find $\mathbf{PR}$.

This is matrix multiplication, so use the `.` operator.

The sizes of these matrices means the calculation is possible.

(f) Find $\mathbf{PQ}$.

Remember to use the `.` operator.

This calculation is not possible due to the sizes of the matrices. Maxima gives an error message:

```
MULTIPLYMATRICES: attempt to multiply nonconformable
matrices.
 -- an error.  To debug this try:  debugmode(true);
```

(g) Find **PS**.

The matrices **P** and **S** have sizes 3×2 and 2×1, respectively, so this is possible and the result is a 3×1 matrix, that is, a column vector.

(h) Find **PT**.

The matrices **P** and **T** have sizes 3×2 and 1×2, respectively, so this product should not be possible. However, Maxima does return an answer!

Maxima does not distinguish between single row matrices (which are sometimes called *row vectors*) and single column matrices (column vectors). It considers them interchangeable as circumstances require. You should take care to multiply only appropriately sized matrices.

(i) Find $\mathbf{R}^2$.

Since **R** is a matrix use the matrix power operator ^^.

(j) Enter

```
R^2;
```

The result is not the same as in part (i).

Here the ^ symbol was used and Maxima squared each individual element of **R**.

(k) Enter

```
P*Q;
```

The result is not **PQ**, since this product does not exist!

Here the * symbol was used and Maxima multiplied corresponding elements of **P** and **Q** together.

If you began studying this subsection from Activity 13 of Unit 9, return to the unit to continue your study.

10.2 Determinants and inverses

The Maxima commands for finding the determinant and inverse of a matrix are demonstrated in the following activity.

Computer activity 54 Finding determinants and matrix inverses

Consider the matrices

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 0 \\ -4 & 4 & 2 \\ -1 & 2 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} 1 & 2 & 0 \\ 5 & 6 & 2 \\ 2 & 2 & 1 \end{pmatrix}.$$

(a) First, enter the matrices $\mathbf{A}$ and $\mathbf{B}$ into Maxima.

(b) Find the determinant of $\mathbf{A}$ by entering

```
determinant(A);
```

> The `determinant` command calculates the determinant of a square matrix.

(c) Find the inverse of $\mathbf{A}$, and assign the result to the variable `C` by entering

```
C:invert(A);
```

> The `invert` command calculates the inverse of a square matrix.
>
> An alternative way to specify the inverse of the matrix `A` in Maxima is `A^^(-1)`, which accords with the notation usually used to denote a matrix inverse.

(d) Check that `C` really is the inverse of `A` by calculating the products `AC` and `CA`.

> Remember to use the symbol `.` for matrix multiplication.
>
> Both products should be the 3×3 identity matrix.

(e) Try to find the inverse of $\mathbf{B}$.

> This gives an error:
>
> ```
> expt:  undefined:  0 to a negative exponent.
>  -- an error.  To debug this try:  debugmode(true);
> ```
>
> This matrix does not have an inverse. You can check this by calculating its determinant.

Another command that is sometimes useful when you are working with matrices is `ident`, which specifies an identity matrix. For example, `ident(2)` gives $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, the 2×2 identity matrix.

Matrix operations

Operation	Syntax	Example
Determinant (of a square matrix)	`determinant(matrix)`	`determinant(A);`
Matrix inverse (of a square matrix)	`invert(matrix)`	`invert(A);`
	or `matrix^^(-1)`	`A^^(-1);`
Identity matrix	`ident(size)`	`ident(2);`

Computer activity 55 Finding determinants and inverses of larger matrices

For each of the following matrices, check whether it is invertible. If it is, find the inverse.

(a) $\begin{pmatrix} 2 & -1 & 1 \\ 2 & 0 & 1 \\ 4 & 2 & 1 \end{pmatrix}$ (b) $\begin{pmatrix} -1 & 0 & 3 & -2 \\ 4 & \frac{1}{2} & -1 & 0 \\ 2 & 0 & 1 & -1 \\ -1 & 2 & 1 & -1 \end{pmatrix}$

If you began studying this subsection from Activity 24 of Unit 9, return to the unit to continue your study.

11 Sequences and series

In this section you will learn how to use Maxima to work with sequences and series.

11.1 Plotting graphs of sequences

In Subsection 3.3 of this *Guide*, you learned how to use the `wxplot2d` command to plot the graphs of algebraic expressions. Here, you will see how to use the same command to plot graphs of sequences. First, in Computer activities 56 and 57, you will learn how to define and work with sequences in Maxima.

Computer activity 56 Working with a sequence specified by a closed form

Consider the sequence (a_n) specified by the following closed form and range of values of n:

$$a_n = 0.8^n \quad (n = 1, 2, 3, \ldots).$$

(a) Define this sequence in Maxima, by entering

```
a[n]:=0.8^n;
```

This command defines the general term a_n of the sequence using the same define operator (`:=`) as is used when defining functions.

Notice that you type the subscript n of the general term within square brackets, but the Maxima output uses a properly formatted subscript.

(b) Find the term a_{100} of the sequence by entering

```
a[100];
```

Notice that when you defined the sequence in Computer activity 56 you did not need to enter the range of values of n. In fact, Maxima places no restriction on the values of n that you can use when calculating terms in a sequence defined by a closed form. For instance, for the sequence (a_n) defined in Computer activity 56 it is possible to calculate values for `a[3.5]` and `a[-4]` using Maxima.

However, these numbers are not terms of the sequence (a_n), which are defined only for integer values of n greater than or equal to 1.

Computer activity 57 Working with a sequence specified by a recurrence relation

Consider the sequence (b_n) specified by the recurrence relation

$$b_1 = 1, \quad b_n = 2b_{n-1} + 1 \quad (n = 2, 3, 4, \ldots).$$

(a) Define this sequence in Maxima by entering the two commands below.

```
b[1]:1;
b[n]:=2*b[n-1]+1;
```

Notice that you assign the value of the first term, `b[1]`, using the `:` operator that is used to assign values to variables, but you specify the recurrence relation using the `:=` operator that is used to define functions.

Unlike for a sequence defined using a closed form, Maxima can calculate a term of a sequence defined using a recurrence relation only when the term number is an integer greater than or equal to the starting value, which in this case is 1.

(b) Find the 5th and 100th terms of this sequence.

(c) Change the value of b_1 to be 10, by entering

```
b[1]:10;
```

and try to find the revised value of b_5.

> Notice that the value of b_5 displayed is unchanged. This is because when Maxima uses sequences, it remembers all the values of the sequence that it has previously calculated, and bases subsequent calculations on them. This improves the efficiency of the calculations. However, it also means that sequences cannot be easily changed.
>
> To change the definition of a sequence, you first need to remove the sequence from Maxima's memory using the `kill` command, and then redefine it.

(d) Revise the sequence (b_n) correctly, by first entering

```
kill(b);
```

and then entering the two commands

```
b[1]:10;
b[n]:=2*b[n-1]+1;
```

to specify the revised sequence.

Now calculate the value of b_5.

The commands for defining and working with sequences are summarised below.

Sequences

Operation	Command	Example
Define a sequence using a closed form	`sequence[n]:=expression in n`	`a[n]:=n^2;`
Define a sequence using a recurrence system	`sequence[first term number]:initial value` `sequence[n]:=expression in sequence[n-1]`	`b[1]:1;` `b[n]:=2*b[n-1];`
Calculate a term in a sequence	`sequence[term number]`	`b[50];`

Note: to change the definition of a sequence, you first have to delete it using `kill`.

In the following activity you will learn how to plot a graph of a sequence. Remember, a graph of a sequence (a_n) is a set of points (n, a_n) plotted for a suitable range of values of n.

Computer activity 58 Plotting the graph of a simple sequence

Plot the graph of the sequence (c_n) specified by

$$c_n = n^2 \quad (n = 1, 2, 3)$$

as follows.

(a) First, define the sequence in Maxima by entering

```
c[n]:=n^2;
```

(b) Create a list of the coordinates of the points to be plotted; that is, the points $(1, c_1)$, $(2, c_2)$ and $(3, c_3)$ and assign it to a variable, say, `pts` (for 'points'), by entering

```
pts:[ [1,c[1]], [2,c[2]], [3,c[3]] ];
```

Notice that you type the coordinates of each point as a list in square brackets. The variable `pts` is a list of lists.

Do not use the variable name `points` for your list of points – this is a word used by Maxima to set plot styles, as you will see next.

(c) Plot the graph of the sequence using the command

```
wxplot2d([discrete, pts], [style, points],
[xlabel, "n"], [ylabel, "c[n]"]);
```

Here, instead of the first argument of the `wxplot2d` command being an expression, or a list of expressions to plot, it is a list containing two elements. The first element is the keyword `discrete`. This tells Maxima that a set of discrete points are to be plotted. The second element is the list of the coordinates of the points to be plotted.

The second argument of the `wxplot2d` command is

```
[style, points].
```

This tells Maxima to simply plot points, and not join them with lines.

The final two arguments specify labels for the x- and y-axes. These arguments were introduced in Subsection 3.3 of this *Guide*.

You can also use other standard arguments of `wxplot2d` that you met earlier. For example, including `[x,0,4]` changes the range of the horizontal axis to $0 \leq x \leq 4$.

The use of `wxplot2d` for plotting sequences is summarised below.

Plotting graphs of sequences

Operation	Command	Example
Plot a sequence	`wxplot2d([discrete, list of coordinate pairs], [style, points], ...)`	`wxplot2d([discrete, [[1,5], [2,10]]], [style, points]);`

Graph plotting options

Option	Argument	Example
Plot points	`[style, points]`	`[style, points]`

In Computer activity 58 the sequence plotted was short, so it was relatively easy to create the list of coordinates to plot. For a longer sequence this can be laborious, but Maxima can help by creating the list for you. This is demonstrated in the next activity.

Computer activity 59 Plotting the graph of a longer sequence

Plot the graph of the sequence given by the recurrence relation

$$d_1 = 0, \quad d_n = \tfrac{1}{2}(2 - d_{n-1}) \quad (n = 2, 3, 4, \ldots, 15)$$

as follows.

(a) First define the sequence (d_n) in Maxima.

(b) To plot the sequence, we need to plot the points

$$(1, d_1),\ (2, d_2),\ (3, d_3),\ \ldots,\ (15, d_{15}).$$

Create a list of these points, and assign the list to the variable `pts`, using the command

```
pts:makelist([n,d[n]], n, 1, 15);
```

The `makelist` command creates a list. The first argument of the command is the general form of the elements of the list. In this case, each element has the form `[n,d[n]]`; that is, it is itself a list containing two elements: the term number and the value of the corresponding term. These are the coordinates of a point to be plotted. Here, the general form is expressed using the dummy variable `n`.

The second argument of the `makelist` command is the name of the dummy variable used, and the third and fourth arguments are the minimum and maximum values taken by the dummy variable.

When you create a long list you may not want to see all the output! Remember that you can end a command with a dollar sign (`$`) rather than a semicolon (`;`) to suppress the display of the output.

(c) Plot the points in the list `pts` by using the command

```
wxplot2d([discrete, pts], [style, points],
 [xlabel, "n"], [ylabel, "d[n]"]);
```

as seen earlier.

The `makelist` command used in the previous activity is summarised below.

Creating lists

Operation	Command	Example
Create a list	`makelist(general term, dummy variable, start value, end value)`	`makelist(2*n,n,1,100);`

As you know from your study of Unit 10, a sequence can have infinitely many terms; such a sequence is called an **infinite sequence**. Since Maxima cannot plot an infinite number of points, when plotting a graph of an infinite sequence, you need to restrict the number of points to a large, but finite, number. Try to choose a number that is large enough to reveal the long-term behaviour of the sequence.

Computer activity 60 Plotting part of some infinite sequences

Use Maxima to plot graphs of the sequences specified below, by restricting the range of values of n to be $n = 1, 2, 3, \ldots, 50$ in each case.

How do you think each sequence will behave as more and more terms are considered?

(a) $r_1 = 1, \quad r_n = 1.5r_{n-1} - 0.1(r_{n-1})^2 \quad (n = 2, 3, 4, \ldots)$

(b) $s_n = n^3 - 10 \quad (n = 1, 2, 3, \ldots)$

(c) $t_n = 1 - 1.1^n \quad (n = 1, 2, 3, \ldots)$

If you began studying this subsection from Activity 17 of Unit 10, return to the unit to continue your study, and in particular learn about the long-term behaviour of sequences.

11.2 Summing series

You can calculate the sums of various series using the Maxima command `sum` which is demonstrated in the following activity.

Computer activity 61 Summing series

(a) Find the sum of the squares of the first 100 integers, that is

$$\sum_{n=1}^{100} n^2,$$

by entering

```
sum(n^2, n, 1, 100);
```

The first argument of the `sum` command is the general term of the series to be summed, which in this case is expressed in terms of the index variable `n`. The second argument is the index variable itself and the third and fourth arguments are the lower and upper limits of the summation.

(b) Write the sum

$$\frac{1}{3} + \left(\frac{1}{3}\right)^2 + \left(\frac{1}{3}\right)^3 + \cdots + \left(\frac{1}{3}\right)^{20}$$

in sigma notation, and then use the Maxima `sum` command to calculate it.

(c) Try to calculate the sum of the infinite series $\sum_{n=1}^{\infty} \frac{1}{4^n}$ by entering

```
sum(1/(4^n), n, 1, inf);
```

Maxima uses `inf` to represent ∞.

Notice that Maxima displays the sum (unevaluated) using sigma notation. By default, Maxima does not attempt to calculate the sums of infinite series.

You can, however, tell Maxima to simplify the result of a summation, which can sometimes result in it finding a value for a sum of an infinite series, if it exists. To do this, you change the value of the system variable `simpsum`, as shown below. (`simpsum` is short for 'simplify sums'.)

(d) Set the system variable `simpsum` to have the value `true` by entering

```
simpsum:true;
```

Now try to calculate $\sum_{n=1}^{\infty} \frac{1}{4^n}$ again.

Maxima now returns the sum of the series.

(e) Find $\sum_{n=1}^{\infty} \frac{1}{n^2}$.

The commands introduced in Computer activity 61 are summarised below.

Summing series

Operation	Command	Example
Sum a series	`sum(general term, index variable, lower limit, upper limit)`	`sum(n^2,n,1,10);`
Simplify summations	`simpsum:true`	`simpsum:true;`

Infinity

Constant	Syntax	Example
∞	inf	`sum(1/n^2,n,1,inf);`

If you began studying this subsection from Activity 32 of Unit 10, return to the unit to continue your study.

12 Taylor polynomials

Taylor polynomials is an optional topic and you will not be assessed on this section.

You could use Maxima to find a Taylor polynomial for a function by using the `diff` command to calculate the required derivatives and then combining these to form the required polynomial. However, Maxima also has a command for calculating Taylor polynomials more efficiently. This is demonstrated in the following activity.

Computer activity 62 Finding Taylor polynomials

(a) Find the cubic Taylor polynomial about $\pi/6$ for the function $f(x) = \cos x$, by entering

```
taylor(cos(x), x, %pi/6, 3);
```

The `taylor` command calculates a Taylor polynomial for the function that is its first argument. Its second argument specifies the input variable of the function.

The third argument of the command is the point about which the Taylor polynomial is to be found, which is also known as the centre of the polynomial. The final argument of the command is the degree of the Taylor polynomial required.

Notice that the Maxima output ends with `...`, which indicates that more terms could have been found. Also, the output line number is followed by `/T/`. This indicates that the output is a (truncated) Taylor series.

(b) Define `p` to be the quintic Taylor polynomial about 0 for the function $f(x) = \tan x$, by entering

```
p(x):=''(taylor(tan(x), x, 0, 5));
```

Here, the `''(  )` command is needed to force Maxima to calculate the Taylor polynomial which is then used to define `p(x)`.

Without the `''(  )` command, `p(x)` would be given by the expression `taylor(tan(x), x, 0, 5)`. So if you tried to find, for example, `p(3)`, then Maxima would interpret this to mean the expression `taylor(tan(3), 3, 0, 5)`; but this is meaningless and would therefore produce an error.

You first met the `''(  )` command in Section 8 of this *Guide*.

(c) Express the Taylor polynomial `p(x)` found above in terms of the variable y by typing

```
p(y);
```

As soon as a Taylor polynomial is used in a further calculation or command, the trailing `...` is dropped, and the Taylor polynomial is converted to a rational expression. This is indicated by the marker `/R/` following the output line number.

(d) Find the decimal value of `p(x)` when $x = 0.1$. Also find the difference between `p(x)` and $\tan x$ at $x = 0.1$.

Maxima warns you it has replaced certain decimal numbers with rational approximations when calculating these values.

The `taylor` command is summarised below.

Taylor polynomials

Operation	Command	Example
Find a Taylor polynomial	`taylor(function,variable, centre,degree)`	`taylor(sin(x),x,0,3);`

Computer activity 63 Plotting Taylor polynomials

Consider the function $f(x) = e^{\sin(x^2)}$.

Plot a graph of the function and the graph of its quintic Taylor polynomial about $x = 1$ on the same axes over the range $0 \leq x \leq 2$.

(Remember, you learnt how to plot more than one function on the same graph in Subsection 3.3 of this *Guide*.)

If you began studying this section from Activity 21 of Unit 11, return to the unit to continue your study.

13 Complex numbers

In this section you will learn how to use Maxima to work with complex numbers.

13.1 Working with complex numbers

The imaginary number i is represented in Maxima by `%i`. You saw this in Subsection 3.2 of this *Guide*, when you solved equations that have complex solutions as well as real solutions. The usual arithmetic operations work with complex numbers in Maxima, and there are commands for finding the real and imaginary parts of a complex number, its conjugate, its modulus and principal argument, as demonstrated in the following activity.

Computer activity 64 Using complex numbers

(a) Assign the complex number $3 + 4i$ to the variable `a`, by entering

```
a:3+4*%i;
```

When Maxima displays a complex number, it usually arranges it with the term in i as the first term, unless the imaginary part is negative and the real part is positive.

(b) Similarly, assign the value $2 - 2i$ to `b`.

(c) Calculate `a+b`, `a-b`, `a*b` and `a/b`.

> The sum and difference are displayed in simplified form, but the product and quotient are not.
>
> To simplify the result of a calculation and display it in Cartesian form (or **rectangular form**, as Maxima calls it) use the command `rectform`.

(d) Find `a*b` in Cartesian form, by entering

```
rectform(a*b);
```

Find `a/b` in Cartesian form similarly.

(e) Find the real and imaginary parts of `a` by entering

```
realpart(a);
```

and

```
imagpart(a);
```

(f) Find the conjugate of `a` by entering

```
conjugate(a);
```

(g) Find the modulus and the principal argument of `b` by entering

```
abs(b);
```

and

```
carg(b);
```

respectively.

> Notice that the `abs` command is the same command that you used earlier to find the magnitude (also called the modulus, or the absolute value) of a real number.
>
> The name of the `carg` command is short for 'complex argument'. The argument calculated is the principal argument, that is, the argument in the interval $(-\pi, \pi]$.

As you saw in Unit 12, any non-zero complex number can be expressed in **polar form** $r(\cos\theta + i\sin\theta)$, where r is the modulus and θ is one of its arguments. The next activity uses this form.

Computer activity 65 Working with the polar form of complex numbers

(a) Assign the complex numbers $3(\cos(5\pi/13) + i\sin(5\pi/13))$ and $4(\cos(7\pi/13) + i\sin(7\pi/13))$ to the variables `c` and `d`, respectively.

(b) Use Maxima to find the moduli and principal arguments of `c*d` and `c/d`.

Hint: use the `trigreduce` command to simplify the initial answers given.

The commands introduced in the activities above are summarised below.

Working with complex numbers

Operation	Command	Example
Real part of a complex number	`realpart( )`	`realpart(1+2*%i);`
Imaginary part of a complex number	`imagpart( )`	`imagpart(1+2*%i);`
Conjugate of a complex number	`conjugate( )`	`conjugate(1+2*%i);`
Modulus of a complex number	`abs( )`	`abs(1+2*%i);`
Principal argument of a complex number	`carg( )`	`carg(1+2*%i);`
Express a complex number in Cartesian form	`rectform( )`	`rectform(2/%i);`

If you began studying this subsection from Activity 27 of Unit 12, return to the unit to continue your study.

13.2 Solving polynomial equations

In Subsection 3.2 of this *Guide* you met the `solve` command that is used to find solutions of equations. This command tries to find all the solutions of the equation (if they can be found using algebraic manipulation), including solutions that are non-real complex numbers. You might remember that in Computer activity 20(e) you solved an equation with non-real solutions.

Computer activity 66 Solving polynomial equations

(a) Use the `solve` command to find the solutions of the equation

$$z^3 + z + 1 = 0,$$

and assign the result to the variable `solns`.

Notice that the solutions are given exactly, although in a rather unwieldy form!

It is often more useful to find decimal approximations to solutions, expressed in Cartesian form, which you can do using the `float` and `rectform` commands.

(b) Express the solutions found in part (a) in Cartesian form, with each real number, and the real and imaginary parts of each complex number, correct to two decimal places.

(c) Find, in Cartesian form, all the solutions of

$$z^4 - 20z^3 + 171z^2 - 626z + 962 = 0.$$

You may have noticed that in Computer activity 66 all the non-real complex solutions occur in complex conjugate pairs. In other words, for each of the equations, if a particular non-real complex number z is a solution, then so is its complex conjugate $\overline{z}$. You saw in Unit 12 that this property holds for any quadratic equation that has real coefficients. In fact, it holds for any polynomial equation that has real coefficients. It doesn't hold for polynomial equations that don't have real coefficients, however. For example, the solutions of the polynomial equation

$$z^2 - 2i = 0$$

are $1 + i$ and $-1 - i$, which are not complex conjugates of each other.

Plotting solutions in the complex plane

It is sometimes informative to plot the solutions of an equation in the complex plane. You can do this in Maxima using the methods that you learned in Subsection 11.1 of this *Guide* for plotting the discrete points that form the graph of a sequence.

To plot the solutions of an equation, you first need to create a list of the points to be plotted, with the first coordinate of each point being the real part of a solution, and the second coordinate being the corresponding imaginary part.

The following activity shows how you can do this.

Computer activity 67 Plotting the solutions of $z^4 + 4z + 5 = 0$

(a) Solve the equation

$$z^4 + 4z + 5 = 0$$

and assign the list of solutions to the variable `s`.

Each element of the list `s` is an equation of the form `z=`*`value`*, where *`value`* is a solution of the equation.

(b) Create a second list containing just the solutions of the equation (without the `z=`) and assign it to the variable `v`, by entering

```
v:makelist(rhs(s[k]), k, 1, length(s));
```

The kth element of the first list `s` is `s[k]`, so `rhs(s[k])` is the right-hand side of this element, which is the kth solution of the original equation.

You met the `makelist` command in Subsection 11.1 of this *Guide*. The index variable `k` starts with the value 1 and ends with the value given by `length(s)`, which is the number of elements in list `s`. This ensures that all of the the solutions of the equation are included in the list `v`.

(c) Create a list of the coordinates of the points to be plotted and assign it to the variable `pts`, by entering

```
pts:makelist([realpart(v[k]), imagpart(v[k])],
 k,1,length(s));
```

The kth solution of the original equation is `v[k]`. The point in the complex plane corresponding to this solution has coordinates

$$(\mathrm{Re}(\texttt{v[k]}), \mathrm{Im}(\texttt{v[k]}))\,,$$

which can be expressed in Maxima syntax as

```
[realpart(v[k]), imagpart(v[k])].
```

The command above uses `makelist` to create a list of such points, one for each solution of the equation.

(d) Finally, plot the points found by entering

```
wxplot2d([discrete,pts], [style,points],
 [xlabel,""], [ylabel,""]);
```

This command plots a set of discrete points. The first two arguments of this `wxplot2d` command are the options that you used to plot graphs of sequences in Subsection 11.1 of this *Guide*. The final two arguments specify that the axis labels are to be left blank.

Notice that since the complex solutions of polynomial equations with real coefficients occur in conjugate pairs, the points plotted are symmetric about the horizontal axis.

Computer activity 67 introduced the `length` command, which is summarised below.

Length of a list

Operation	Command	Example
Find the length of a list	`length(list)`	`length(A);`

Computer activity 68 Finding and plotting the seventh roots of unity

(a) Find the seventh roots of unity by solving the equation

$$z^7 = 1.$$

Give the solutions in Cartesian form, with each real number, and the real and imaginary parts of each complex number, correct to two decimal places.

(b) Plot the solutions in the complex plane.

If you began studying this subsection from Activity 35 of Unit 12, return to the unit to continue your study.

14 Maxima accessibility guide

This section outlines issues associated with the use of Maxima if you interact with a computer using only a keyboard, use a screenreader or require different colour settings. This information should help you decide whether the wxMaxima interface is suitable for you, or whether you should consider using an alternative interface, such as the command-line interface. It also explains how to configure Maxima to make it easier for you to use and contains alternative versions of Subsection 1.2 and parts of Section 2 describing how to use the command-line interface.

Activities within this section are numbered to match corresponding activities in Sections 1 and 2. If you decide to use the command-line interface, the format of the solutions given at the end of this *Guide* may not exactly match the output you obtain.

14.1 Choosing and configuring your interface

This subsection describes how wxMaxima can be configured to help those with additional requirements. If you find these changes are not sufficient, you may wish to use the command-line interface to Maxima instead.

Using wxMaxima with a keyboard alone

The majority of the wxMaxima interface is accessible using a keyboard alone. On a Windows computer you can access the wxMaxima menus by pressing and releasing the Alt key, navigating through the menus using the keyboard arrow keys, and then pressing Enter to make your selection. Alternatively, you can press and hold down the Alt key, and then press the keyboard key that corresponds to the underlined letter in a menu entry to select it. In both cases, you can leave menu navigation without making a selection by pressing the Esc key.

On an Apple Mac you can access the wxMaxima menus by pressing and releasing the F2 key while holding down the Ctrl and Fn keys (or just Ctrl on some keyboards), navigating through the menus using the keyboard arrow keys, and then pressing Enter to make your selection. You can leave menu navigation without making a selection by pressing the Esc key.

The wxMaxima toolbar cannot be accessed using the keyboard alone, but this is not used extensively in this module and alternative typed commands for accessing the functions are given as required.

Changing wxMaxima colours and fonts

You can change the appearance of the text used by wxMaxima as follows. First, click on the following toolbar icon.

Alternatively, select `Configure` from the `Edit` menu (or, on an Apple Mac, select `Preferences` from the `wxMaxima` menu).

In the configuration window that opens, select the following 'Style' icon.

You can then change the colour, style, and sometimes the font and size, of individual elements of the wxMaxima interface by first choosing the appropriate element from the 'Styles' list, and then selecting your preferred settings.

Where a selected element has a greyed-out 'Choose font' button this is usually because it shares its font with other elements of the interface. For example, those elements that form part of the mathematical output of the interface share a font that you can change by first clicking on the Math font 'Choose font' button and then selecting your preferred font and size. Most of the remaining elements with a greyed-out font button share a font that you can change in a similar way by clicking on the Default font 'Choose font' button. If you choose a large font size, some horizontal scrolling may be necessary to read all the text displayed in a worksheet.

Unfortunately, the Configuration window is not keyboard accessible, and so if you only use a keyboard you may need the assistance of a non-medical helper to initially configure the system.

The Configuration window does not change the colours and fonts used in graphs produced by Maxima. Subsection 14.7 explains how to change the colours, fonts and line thickness used in graphs. You should read Subsection 14.7 in conjunction with Subsection 3.3, which discusses how to plot graphs.

Using a screen magnifier or screenreader

The text in wxMaxima can become pixelated when you use a screen magnifier. An alternative is to increase the size of the text font as described above and/or use the zooming facilities on the `View` menu, which you can also access using `Alt-I` (zoom in) and `Alt-O` (zoom out).

The wxMaxima interface is not accessible to a screenreader. If you use a screenreader you might wish to consider using the command-line interface to Maxima instead of the wxMaxima interface. You should ensure your screenreader is set to read all the punctuation. Details of using Maxima with the command-line interface are given below. The graphs produced by Maxima are not accessible to a screenreader. You might need assistance when using these.

14.2 Starting, testing and configuring Maxima using the command-line interface

This subsection describes how to start, test and configure Maxima when using the command-line interface. If you are using the command-line interface you should read this subsection instead of Subsection 1.2, which this replaces.

Computer activity 2 Starting Maxima

Start the command-line interface to Maxima on your computer and keep it open to work with as you study this section.

If you cannot remember how to do this on your computer, see the OU Maxima website: `learn1.open.ac.uk/site/maxima`.

The command-line interface to Maxima is illustrated in Figure 12.

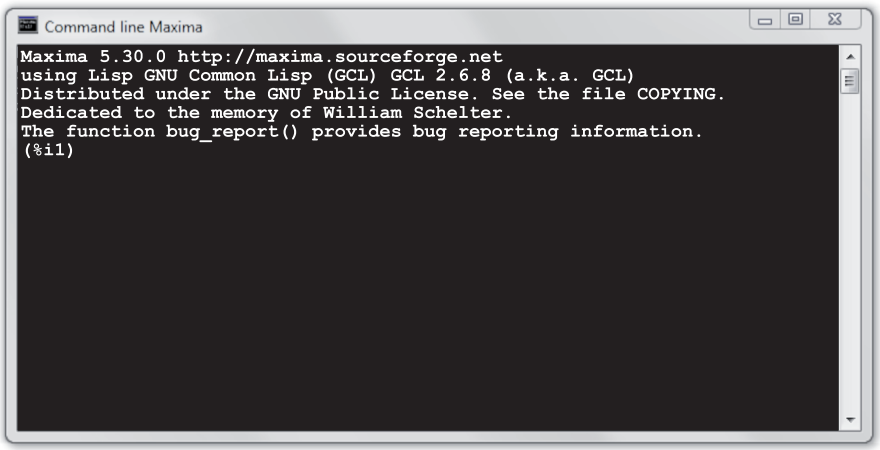


Figure 12 The initial command-line interface window

At the top of the window is some introductory text, followed by

`(%i1)`

This is how Maxima labels lines. The `i` stands for 'input': this is input line number 1 and is where you can type your first command. These commands instruct Maxima to either perform calculations or change system settings.

The percentage symbol (`%`) is used by Maxima to indicate objects built into the system. There is more about this later.

The following activity shows you how to use Maxima to perform a simple calculation and checks that your system is working properly. Follow the steps using Maxima as you read the activity.

If you encounter unexpected problems when working through this activity, check the Frequently asked questions (FAQs) section on the OU Maxima website: `learn1.open.ac.uk/site/maxima`.

Computer activity 3 Your first Maxima calculation

Follow these steps to use Maxima to calculate $2 + 3 \times \frac{4}{5}$.

(a) Click anywhere on the Maxima window, to ensure that the text you type next is directed to the correct place.

(b) Type the following exactly as it appears here:

```
2+3*4/5;
```

including the semicolon (`;`) at the end.

> The text you type will appear next to the input line number `(%i1)`, as indicated by a flashing horizontal line known as the **editing cursor**.

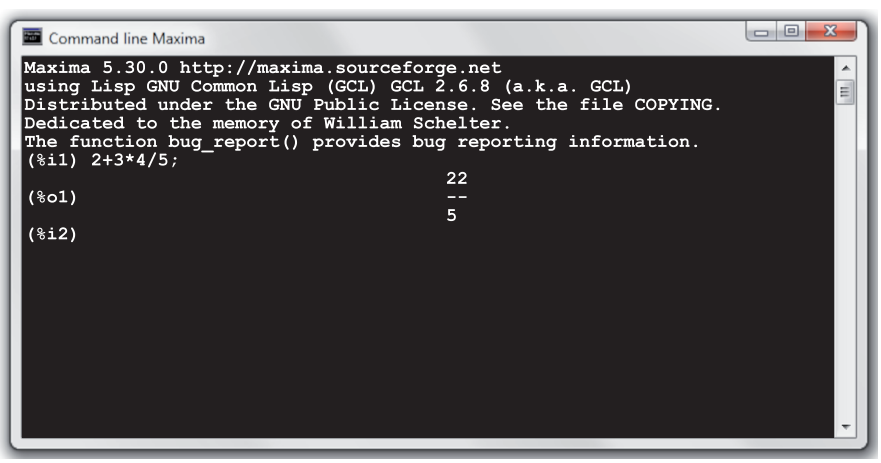
(c) Press Enter.

If you forget to type the semicolon (;) at the end of the command, the editing cursor simply moves to the next line, waiting for you to type more. Maxima will continue to do this until you indicate you have reached the end of the command by typing a semicolon. So if you forgot the semicolon, then type it now and press Enter again.

Pressing Enter instructs Maxima to calculate the expression. This is known as **evaluating** the expression.

The result is displayed under the input line, preceded by the label `(%o1)`. The `o` stands for 'output', so this label identifies the line as output line 1. It gives the output corresponding to input line 1. Throughout each Maxima session the line numbering starts at 1 and then increases by one for each new input (and output).

Maxima then displays another input line number, ready for you to enter your next command.



If, in Computer activity 3, Maxima behaved as described, then you have a working system to use throughout MST124.

The result of the calculation in Computer activity 3 was displayed as formatted mathematics over several lines, replicating how mathematics is usually written. This can, however, cause problems if you are using a screenreader. The display behaviour can be changed for every Maxima session as described in the following activity.

Computer activity 4 Configuring the Maxima display of mathematics

You should only do this activity if you wish to change how Maxima displays mathematics.

(a) First exit Maxima by entering the command

```
quit();
```

(on some installations you may also have to close the window).

(b) Turn off the two-dimensional display of mathematics by adding the Maxima command `display2d:false;` to your `maxima-init.mac` configuration file, as follows.

(i) Follow the instructions in the Accessibility section of the OU Maxima website `learn1.open.ac.uk/site/maxima` to locate your `maxima-init.mac` configuration file, or create one in the correct location as appropriate.

(ii) Open your `maxima-init.mac` file in a plain text editor (such as Notepad on a Windows computer, or TextEdit on an Apple Mac), and add the text

```
display2d:false;
```

on a new line. There are further details on how to do this in the Accessibility section of the OU Maxima website.

(iii) Save your modified `maxima-init.mac`, and close your text editor.

(c) Restart Maxima and enter the calculation from Computer activity 3 again, that is, enter

```
2+3*4/5;
```

Notice that Maxima now displays the result on a single line:

```
22/5
```

Another useful configuration change is to make Maxima display the result of a command left-justified against the output line number, instead of the default behaviour of displaying it centred on the line. This can be achieved by entering the command

```
leftjust:true;
```

either in the Maxima window, or, if you wish to make the change permanent, in your `maxima-init.mac` file as described above.

Closing wxMaxima

When you have finished using Maxima, close it by using one of the following methods.

- Enter `quit();`
- On a Windows computer, click the usual small cross button at the top right-hand corner of the Maxima window.

14.3 Troubleshooting Maxima using the command-line interface

If you are using the command-line interface to Maxima and it does not seem to be responding, try the following suggestions.

Troubleshooting Maxima

- Check you have ended your previous command with either a semicolon (`;`) or a dollar (`$`). Maxima will not do anything without them!
- Check that Maxima is not waiting for you to type something. Is a question being displayed that you have not yet answered?
- If Maxima is taking too long to evaluate a command, you can interrupt it by pressing `Ctrl-C`.

  If you do this, you may see the following output, or something similar, which you can safely ignore.

```
Maxima encountered a Lisp error:

  EXT:GC: User break

Automatically continuing.

To enable the Lisp debugger set *debugger-hook* to nil.
```

Note that pressing `Ctrl-C` at other times may close your Maxima session.

If you encounter other problems when using Maxima, check the Frequently asked questions (FAQs) section of the OU Maxima website: `learn1.open.ac.uk/site/maxima`.

14.4 Reusing and editing commands in the command-line interface

In Computer activity 8 you may have entered the expression twice: once to evaluate it exactly and once for a decimal answer. This is, however, not necessary. When using the command-line interface, you can edit and re-evaluate previously entered commands or calculations, as shown in the following activity.

Computer activity 10 Editing a command

(a) Calculate $2\sqrt{3}+5\sqrt{27}$ by entering the following line into Maxima.

```
2*sqrt(3)+5*sqrt(27);
```

(b) Edit your entry in part (a) to calculate $2\sqrt{3} - 5\sqrt{27}$ as follows.

(i) Press the keyboard up arrow key once, so that the previously entered command appears against the current input line number (on an Apple Mac you may need to hold down the Alt key while pressing the keyboard up arrow key once).

> You can access all the commands you have previously entered in your Maxima session using the up and down keyboard arrow keys.

(ii) Edit the expression by moving the editing cursor using the left and right keyboard arrow keys, typing new text and using the ← (backspace) key to make deletions.

(iii) Press Enter.

Maxima also permits the result of one calculation to be used in another calculation. The symbol `%` represents the result of the last evaluated command. So, for example, if you enter `sqrt(%);` Maxima calculates the square root of the last evaluated output. You can also use the output (or input) of other lines, by typing the line label in a calculation. For example, if you enter `%o5^3;` Maxima calculates the cube of the expression in output line 5.

Computer activity 11 Using the results of previous calculations

(a) Enter `6*2;` into Maxima and evaluate it.

(b) On the following line type `3+%;` press Enter.

> Maxima returns `15`, the sum of 3 and the previous answer.

(c) Enter `%o` `+ 5;` with ▢ replaced by the output line number corresponding to the result of `6*2;` entered earlier.

> This command adds 5 to the result of the line referenced.

14.5 Saving and printing your work using the command-line interface

After working with Maxima you will probably want to save your calculations so that you can reuse or read them at a later date.

When you are using the command-line interface you cannot save the session directly, as you do when saving other files. Instead there are two stages in saving your work:

- If you want to keep a record of the contents of the command-line window, then you need to save a *transcript* of the session. You can then print this transcript if needed.
- If you want to save the values of any variables and functions you have defined, so that you can continue working with them in a new Maxima session, then you will need to save the current state of the system. Note that this does not give you access to the commands used to define the variables or functions, so you cannot edit them in the new session.

These two processes are illustrated in Computer activity 16.

When saving your work, you need to tell Maxima exactly where to save the file containing the session state or your transcript by giving the complete *pathname* of the file, which includes not only the name of the file but also the hierarchy of folders in which the file is located. Unless you do this, Maxima will try to save the file in the location in which the system is installed on your computer. Not only is this not recommended, but as a computer user, you may not have sufficient permission to do this.

The pathname of a file depends on its location, the type of computer you are using, and your computer username. In the following activity, we assume that a user with the computer username 'bill' wishes to save a transcript of his Maxima session in a file called `transcript.txt` within a folder called `maxima` which he created in Computer activity 1. The `maxima` folder is within Bill's 'Documents' area on a recent Windows computer (that is, Windows Vista, 7, 8 or 10). In this case, the full pathname of his file is

```
C:/Users/bill/Documents/maxima/transcript.txt
```

(Notice that Maxima separates parts of the pathname with a forward slash (`/`) rather than the usual Windows backward slash (\).)

If Bill were using an older Windows computer (such as Windows XP), the equivalent pathname would be

```
C:/Documents and Settings/bill/Documents/maxima/transcript.txt
```

Using an Apple Mac the equivalent is

```
/Users/bill/Documents/maxima/transcript.txt
```

Computer activity 16 Saving your work

(a) Create a transcript of your current session by following these steps.

(i) Enter
`writefile("C:/Users/bill/Documents/maxima/transcript.txt");`
replacing the pathname with one appropriate to you and your system.

The path of the file is given in double quote marks.

The command tells Maxima to save a copy of all the subsequent input and outline lines to the file named `transcript.txt` in your chosen folder.

The output

```
#<OUTPUT BUFFERED FILE-STREAM CHARACTER
C:\Users\bill\Documents\maxima\transcript.txt>
```

done

is given, where the pathname is the one you gave in the command.

(ii) Enter

```
playback();
```

This command redisplays each of the input and output lines of your session, and when used in conjunction with the `writefile` command results in a copy of these lines being stored in your chosen file.

To playback only a selection of the lines from your session, you can use a command of the form

```
playback([first line number, last line number]);
```

which plays back only those lines within the given range.

(iii) Enter

```
closefile();
```

This stops the recording of the session into the file.

The output

```
#<CLOSED OUTPUT BUFFERED FILE-STREAM CHARACTER
C:\Users\bill\Documents\maxima\transcript.txt>
```

done

is given.

(iv) Find the file `transcript.txt` on your computer and open it using a text editor, for example, Microsoft Word.

The file contains a copy of the input and output lines from your session. (When opened in some text editors the file may appear all on one long line.)

(b) Save your current session then reload it, by following these steps.

(i) Enter
`save("C:/Users/bill/Documents/maxima/mysession.lisp", all);`
replacing the pathname with one appropriate to you and your system.

This tells Maxima to save your current session to the file `mysession.lisp` in your chosen folder. (Lisp is the computer programming language used to create Maxima, and the `save` command saves your work using commands from this language.)

The path of the file is output.

(ii) Close Maxima by entering

```
quit();
```

(iii) Restart Maxima in the usual way.

(iv) Load your previously saved session by entering
`load("C:/Users/bill/Documents/maxima/mysession.lisp");`
using the same pathname you used in part (b)(i).

This tells Maxima to restore the state of the system to that saved in the specified file.

(v) Enter

```
values;
```

to display the names of all known variables.

You should see that all the variables from your previous session, if any, are known. You might like to check their values are as expected too.

There is more about variables and the `values` command in Subsection 2.6.

An alternative way of using the `writefile` and `closefile` commands is to enter the first at the very start of your session and the second at the end, so that everything you do is recorded.

Saving your work

Operation	Command
Start writing a transcript	`writefile("file path");`
Stop writing a transcript	`closefile();`
Playback all the input and output in a session	`playback();`
Playback a range of input and output	`playback([start line number, end line number]);`
Save the state of a session	`save("file path", all);`
Reload a session	`load("file path");`

You can also obtain an image of the Maxima window by taking a 'screenshot'. To do this on a Microsoft Windows computer, first click on the Maxima window, then press `Alt-PrintScreen` (that is, hold down the Alt keyboard key while pressing the `PrintScreen` key, which may be labelled PrtScr, Prt Scrn or something similar). This stores the image of the Maxima window, which you can then paste into a suitable application. If you are using Windows Vista, Windows 7, 8 or 10, you might like to use the 'Snipping Tool', available by typing 'snip' into the Start menu.

To take a screenshot on an Apple Mac computer, press `Cmd-Ctrl-Shift-4` (that is, hold down the Command, Ctrl and ⇧ keyboard keys while pressing the 4 key). Next press the space bar, move the camera pointer over the Maxima window and click. This stores the image of the Maxima window, which you can then paste into a suitable application. You may also like to use the Grab application from the Utilities folder.

14.6 Plotting graphs without wxMaxima

When using the command-line interface, you should use the `plot2d` command rather than the `wxplot2d` command described in Section 3.3. These commands work similarly, and have the same arguments. The main difference is that `plot2d` opens a new computer window containing the graph.

The graph window can be closed by pressing the usual cross at the top right-hand corner, or by pressing `Alt-F4`. You will not be able to enter any commands into the Maxima command-line interface until the graphing window is closed.

The graph can be saved as a `.png`, `.pdf` or `.svg` file using the save button. These files can be opened with many image viewing programs and can be inserted into word-processed documents.

14.7 Changing properties of graphs

You can change the thickness of curves plotted in graphs by including an additional argument in the `wxplot2d` command, of the form

```
[style, [lines, 3]]
```

where the number indicates how thick the line should be. By default, lines have thickness 1.

So, to plot a graph of the equation $y = 2x$ with a thicker line, enter, for example,

```
wxplot2d(2*x, [x, 0, 1], [style, [lines, 3]]);
```

Typing this additional argument each time you want to plot a graph can be cumbersome, so you might like to assign this argument to a variable that you can reuse. For example, to assign your preferred style to the variable `thick` enter

```
thick:  [style, [lines, 3]];
```

Then you can plot graphs using commands such as

```
wxplot2d(2*x, [x, 0, 1], thick);
```

To change the colour of a curve, you can include an argument of the form `[color, red]` as described in Subsection 3.3. Alternatively, you can add a further element to the styles argument described above. For example, including

```
[style, [lines, 3, 5]]
```

produces curves with thickness 3 and colour number 5, which is black. The colour number codes used by the `style` argument are given in the table below.

`style` argument colour numbers

Number	Colour
1	blue
2	red
3	green
4	magenta
5	black
6	cyan

Note that the value of any variable you define to help set curve properties (such as `thick` above) will be lost when you close Maxima at the end of a session unless you save your session. The `style` argument also only changes properties of the curve plotted, not other aspects of the graph such as the axes and labels.

Changing the axes and labels, and ensuring such changes are automatically used each time Maxima is started, is more complicated. A computer file is provided in the Accessibility section of the OU Maxima website `learn1.open.ac.uk/site/maxima` which you can download and configure to set the graph properties to your needs. These settings will then be automatically used for each graph in every Maxima session. Details of how to download and use this are given in the Accessibility section of the OU Maxima website.

Computer methods for CAS Activities in Books A–D

Computer solution to Unit 5, Activity 17

First, load the `implicit_plot` package and set the default plotting options to use equal scales on the x- and y-axes.

```
(%i1) load(implicit_plot);
(%o1) C:\maxima-5.38.1\share\maxima\5.38.1_5\share\contrib\
      implicit_plot.lisp
(%i2) set_plot_option(same_xy)$
```

This line is ended with `$` to prevent the display of the output of the command.

(a) Assign the equation representing the parabola and the equation representing the circle to the variables `p` and `c`, respectively.

```
(%i3) p:y=2*x^2-10*x+14;
```

(%o3) $y = 2x^2 - 10x + 14$

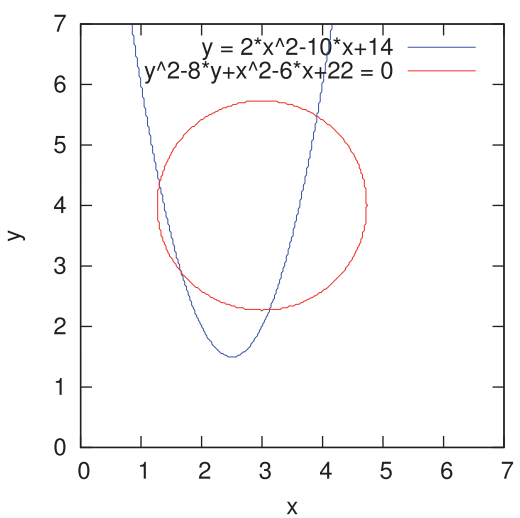
```
(%i4) c:x^2-6*x+y^2-8*y+22=0;
```

(%o4) $y^2 - 8y + x^2 - 6x + 22 = 0$

Now plot the two curves. Appropriate axis ranges are chosen to ensure the whole of the circle is shown.

```
(%i5) wximplicit_plot([p,c], [x,0,7], [y,0,7]);
```

(%t5)



(%o5)

(b) To find the points of intersection of the parabola and circle, we solve the equations representing each simultaneously, for x and y. The equations have previously been assigned to the variables `p` and `c`, respectively.

```
(%i6) solve([p,c], [x,y]);
```

(%o6) [[x = 1.304970075461879, y = 4.356193142057383],
[x = 1.664051355206847, y = 2.897620365246265],
[x = 3.121372031662269, y = 2.272207016670814],
[x = 3.909606299212598, y = 5.473980309423348]]

So, to 2 d.p., the four points of intersection are

$$(1.30, 4.36), (1.66, 2.90), (3.12, 2.27) \text{ and } (3.91, 5.47).$$

(c) The maximum number of intersections of a circle and a parabola is four. This fact can be justified by the geometric argument that each 'half' of the parabola (each part on one side of the vertex) can meet the circle at most twice.

Alternatively, if the equation of the parabola is used to substitute for y in the equation of the circle, then an equation for x containing terms in x^4, x^3, x^2, x and a constant term is obtained, and such an equation can have at most four solutions (by a result that you will meet in Unit 12).

Computer solution to Unit 7, Activity 23

(a) The function f is defined as follows.

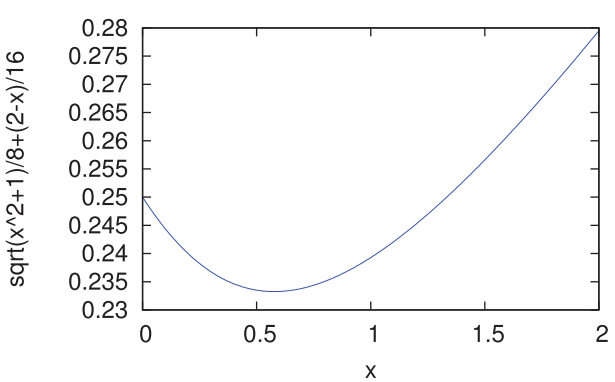
```
(%i1) f(x):=(sqrt(1+x^2))/8+(2-x)/16;
```

(%o1) $\mathtt{f(x)} := \dfrac{\sqrt{1+\mathtt{x}^2}}{8} + \dfrac{2-\mathtt{x}}{16}$;

(b) The graph of $f(x)$ is plotted below.

```
(%i2) wxplot2d(f(x),[x,0,2]);
```

(%t2)



(%o2)

(c) The value of x that gives the minimum value of $f(x)$ seems to be about 0.6.

(d) The derivative of f can be found and assigned to `df(x)` as follows.

```
(%i3) df(x):=''(diff(f(x),x));
```

(%o3) $\mathtt{df(x)} := \dfrac{\mathtt{x}}{8\sqrt{\mathtt{x}^2+1}} - \dfrac{1}{16}$

So the derivative of f is $f'(x) = \dfrac{x}{8\sqrt{x^2+1}} - \dfrac{1}{16}$.

(e) First, we try to use the `solve` command to find the solution of `df(x)=0`.

```
(%i4)  solve(df(x)=0);
```

$$(\%o4)\quad [x = \frac{\sqrt{x^2+1}}{2}]$$

Maxima returns a rearranged version of the equation, but does not solve it. The `solve` command cannot find an exact solution of equation `df(x)=0`, so we look for an approximation to the solution instead.

To do this we need to find an interval containing the solution. We know from the graph of $f(x)$ that its gradient is negative to the left of the minimum, and in particular at $x = 0$, so `df(0)`< 0. Also, the gradient of $f(x)$ is positive to the right of the minimum, in particular at $x = 1$, so `df(1)`> 0. So a solution of `df(x)=0` must lie in the interval $[0, 1]$. An alternative way to find this interval is to plot a graph of `df(x)`.

```
(%i5)  find_root(df(x),x,0,1);
(%o5)  0.5773502691896258
```

The solution of the equation $f'(x) = 0$ is $x = 0.577$ (to 3 s.f.).

(f) The man should join the road 0.58 km (to the nearest 10 metres) along from the point that is closest to his initial position.

(Because the graph of f indicates that the minimum value of $f(x)$ occurs at a local minimum rather than at one of the endpoints of the interval $[0, 2]$, in the solution above we don't consider the values of $f(0)$ and $f(2)$.)

Computer solution to Unit 8, Activity 48

(a) The domain of f is the interval $[-1, \infty)$.

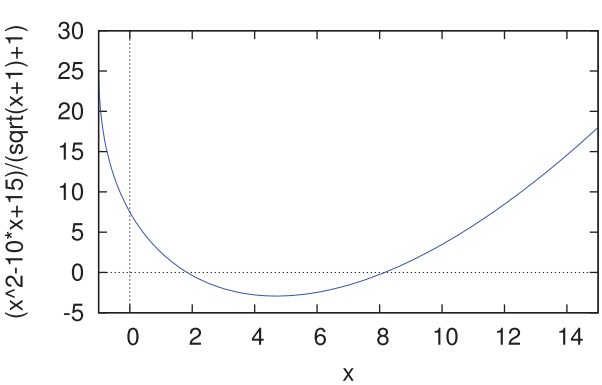
(b) The graph of f can be plotted as follows.

```
(%i1)  f(x):=(x^2-10*x+15)/(1+sqrt(x+1));
```

$$(\%o1)\quad f(x) := \frac{x^2 - 10x + 15}{1 + \sqrt{x+1}};$$

```
(%i2)  wxplot2d(f(x), [x,-1,15]);
```

(%t2)



(%o2)

(c) The x-intercepts of f can be found by solving $f(x) = 0$.

First, solve the equation, assigning the result to the variable `solns`.

```
(%i3) solns:solve(f(x)=0);
```

(%o3) $[x = 5 - \sqrt{10}, x = \sqrt{10} + 5]$

Next, extract the two solutions from the list and assign them to the variables `sol1` and `sol2`, respectively.

```
(%i4) sol1:rhs(solns[1]);
```

(%o4) $5 - \sqrt{10}$

```
(%i5) sol2:rhs(solns[2]);
```

(%o5) $\sqrt{10} + 5$

Finally, express the solutions as decimal approximations.

```
(%i6) float(sol1);
```

(%o6) 1.83772233983162

```
(%i7) float(sol2);
```

(%o7) 8.16227766016838

So the two x-intercepts of f are 1.84 and 8.16 (to 3 s.f.).

(d) The area between the graph of f and the x-axis, between the two x-intercepts, is

$$-\int_{1.837\ldots}^{8.162\ldots} \frac{x^2 - 10x + 15}{1 + \sqrt{x+1}}\,\mathrm{d}x.$$

We shall concentrate on evaluating

$$\int_{1.837\ldots}^{8.162\ldots} \frac{x^2 - 10x + 15}{1 + \sqrt{x+1}}\,\mathrm{d}x,$$

that is,

$$\int_{1.837\ldots}^{8.162\ldots} f(x)\,\mathrm{d}x,$$

remembering the required answer is the negative of this integral.

Maxima can integrate this function exactly.

```
(%i8) integrate(f(x),x,sol1,sol2);
```

(%o8) $-\left(450\log\left(\sqrt{\sqrt{10}+6}+1\right) - 450\log\left(\sqrt{6-\sqrt{10}}+1\right) - 15\,10^{\frac{3}{2}} + \sqrt{6-\sqrt{10}}\left(38\sqrt{10}+66\right) + \sqrt{\sqrt{10}+6}\left(38\sqrt{10}-66\right)\right)/15$

```
(%i9) float(%);
```

(%o9) −12.37959499949233

So

$$\int_{1.837...}^{8.162...} f(x)\,\mathrm{d}x = -12.4 \text{ (to 3 s.f.)}.$$

Remembering the required area is the negative of this, the required area is 12.4 (to 3 s.f.).

Computer solution to Unit 8, Activity 49

The area of the cross-section, in square centimetres, is

$$\int_{-1}^{1.5} \tfrac{1}{3}\sqrt{x^3+1}\,dx.$$

The `integrate` command cannot evaluate this definite integral, so we find an approximate value instead. First, assign the integrand to the variable `y`.

```
(%i1) y:(1/3)*sqrt(x^3+1);
```

$$(\%o1)\quad \frac{\sqrt{x^3+1}}{3}$$

Then find an approximate value for the definite integral.

```
(%i2) quad_qags(y,x,-1,1.5);
```

```
(%o2) [0.939242718890935, 7.1021410974481110^-10, 231, 0]
```

The value of the definite integral is 0.939 (to 3 s.f.). So the volume of plastic required to make one metre of the edging is approximately 93.9 cm^3.

Computer solution to Unit 8, Activity 50

The change in the displacement of the object from time $t = 0$ to $t = 10$ is

$$\int_0^{10} \sin\left(\frac{t^2}{150}\right)\,\mathrm{d}t.$$

The `integrate` command calculates this definite integral in terms of values of the Error function, `erf`. Instead, we shall find an approximation to its value using the `quad_qags` command.

```
(%i1) quad_qags(sin(t^2/150),t,0,10);
```

```
(%o1) [2.15266541148388, 2.3899387041435910^-14, 21, 0]
```

So the value of the definite integral is 2.15 (to 2 d.p.).

The displacement of the object at time $t = 10$ is $(6 + 2.15)\,\mathrm{m} = 8.15\,\mathrm{m}$, to the nearest centimetre.

Computer solution to Unit 9, Activity 28

(a) The matrix form of the equations is

$$\begin{pmatrix} 2 & 2 & 3 \\ 2 & 3 & 4 \\ 4 & 7 & 10 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix}.$$

Assign the coefficient matrix to the variable `A`, and the vector formed by the right-hand sides of the equations to `b`.

```
(%i1) A:matrix([2,2,3],[2,3,4],[4,7,10]);
```

$$(\%o1) \quad \begin{bmatrix} 2 & 2 & 3 \\ 2 & 3 & 4 \\ 4 & 7 & 10 \end{bmatrix}$$

```
(%i2) b:matrix([2],[-1],[3]);
```

$$(\%o2) \quad \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$$

The inverse of `A` can be found as follows.

```
(%i3) invert(A);
```

$$(\%o3) \quad \begin{bmatrix} 1 & \frac{1}{2} & -\frac{1}{2} \\ -2 & 4 & -1 \\ 1 & -3 & 1 \end{bmatrix}$$

So we have

$$\begin{pmatrix} 2 & 2 & 3 \\ 2 & 3 & 4 \\ 4 & 7 & 10 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & \frac{1}{2} & -\frac{1}{2} \\ -2 & 4 & -1 \\ 1 & -3 & 1 \end{pmatrix}.$$

Therefore

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 & \frac{1}{2} & -\frac{1}{2} \\ -2 & 4 & -1 \\ 1 & -3 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix}.$$

This can be calculated using

```
(%i4) invert(A).b;
```

$$(\%o4) \quad \begin{bmatrix} 0 \\ -11 \\ 8 \end{bmatrix}$$

The solution is $x = 0$, $y = -11$, $z = 8$.

Note that an alternative way to solve these equations using Maxima is to use the `solve` command, as follows.

```
(%i5) solve([2*x+2*y+3*z=2, 2*x+3*y+4*z=-1, 4*x+7*y+10*z=3], [x,y,z]);
(%o5) [[x = 0, y = -11, z = 8]]
```

(b) The matrix form of the equations is

$$\begin{pmatrix} 3 & 2 & 1 \\ 4 & 3 & 1 \\ 7 & 5 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}.$$

Assign the coefficient matrix to the variable `A`, and the vector formed by the right-hand sides of the equations to `b`.

```
(%i6) A:matrix([3,2,1],[4,3,1],[7,5,1]);
```

$$(\%o6)\quad \begin{bmatrix} 3 & 2 & 1 \\ 4 & 3 & 1 \\ 7 & 5 & 1 \end{bmatrix}$$

```
(%i7) b:matrix([1],[2],[1]);
```

$$(\%o7)\quad \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

The inverse of `A` can be found as follows.

```
(%i8) invert(A);
```

$$(\%o8)\quad \begin{bmatrix} 2 & -3 & 1 \\ -3 & 4 & -1 \\ 1 & 1 & -1 \end{bmatrix}$$

So we have

$$\begin{pmatrix} 3 & 2 & 1 \\ 4 & 3 & 1 \\ 7 & 5 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 2 & -3 & 1 \\ -3 & 4 & -1 \\ 1 & 1 & -1 \end{pmatrix}.$$

Therefore

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 & -3 & 1 \\ -3 & 4 & -1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}.$$

This can be calculated using

```
(%i9) invert(A).b;
```

$$(\%o9)\quad \begin{bmatrix} -3 \\ 4 \\ 2 \end{bmatrix}$$

The solution is $x = -3$, $y = 4$, $z = 2$.

Note that an alternative way to solve these equations using Maxima is to use the `solve` command, as follows.

```
(%i10) solve([3*x+2*y+z=1, 4*x+3*y+z=2, 7*x+5*y+z=1], [x,y,z]);
(%o10) [[x = -3, y = 4, z = 2]]
```

Computer solution to Unit 9, Activity 29

(a) The matrix form of the equations is

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 4 & 7 & 7 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \\ 1 \end{pmatrix}.$$

The coefficient matrix is $\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 4 & 7 & 7 \end{pmatrix}$.

Its determinant can be found using Maxima as follows.

```
(%i1) determinant(matrix([1,2,3],[2,3,1],[4,7,7]));
(%o1) 0
```

The determinant of the coefficient matrix is zero, so the coefficient matrix is non-invertible and the equations do not have a unique solution.

(b) The matrix form of the equations is

$$\begin{pmatrix} 2 & 2 & 3 \\ 2 & 3 & 1 \\ 4 & 7 & 7 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \\ 1 \end{pmatrix}.$$

The coefficient matrix is $\begin{pmatrix} 2 & 2 & 3 \\ 2 & 3 & 1 \\ 4 & 7 & 7 \end{pmatrix}$.

Its determinant can be found using Maxima as follows.

```
(%i2) determinant(matrix([2,2,3],[2,3,1],[4,7,7]));
(%o2) 14
```

The determinant of the coefficient matrix is 14. This is non-zero, so the coefficient matrix is invertible and the equations have a unique solution.

(c) The matrix form of the equations is

$$\begin{pmatrix} 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ 2 & 0 & 3 & -1 \\ 6 & 2 & 4 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ w \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 1 \\ 0 \end{pmatrix}.$$

The coefficient matrix is $\begin{pmatrix} 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ 2 & 0 & 3 & -1 \\ 6 & 2 & 4 & 0 \end{pmatrix}$.

Its determinant can be found using Maxima as follows.

```
(%i3) determinant(matrix([1,1,-1,1],[1,-1,1,1],[2,0,3,-1],[6,2,4,0]));
(%o3) 0
```

The determinant of the coefficient matrix is zero, so the coefficient matrix is non-invertible and the equations do not have a unique solution.

Computer solution to Unit 10, Activity 20

(a) $x_n = -\frac{1}{3}(1.2)^n \quad (n = 1, 2, 3, \ldots)$.

The first 30 points on the graph of the sequence (x_n) can be plotted as follows.

First, define the sequence.

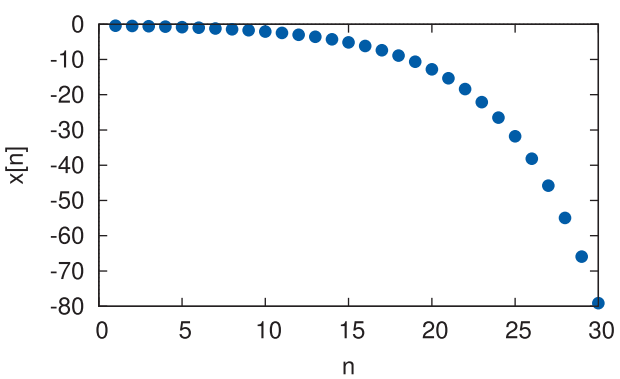
```
(%i1) x[n]:=-(1/3)*(1.2)^n;
```

(%o1) $x_n := \left(-\frac{1}{3}\right) 1.2^n$

Then generate the points to be plotted and plot them.

```
(%i2) xpts:makelist([n,x[n]],n,1,30)$
(%i3) wxplot2d([discrete,xpts], [style,points],
       [xlabel,"n"], [ylabel,"x[n]"]);
```

(%t3)



(%o3)

The sequence (x_n) is decreasing and $x_n \to -\infty$ as $n \to \infty$.

(b) $y_n = 5(-0.9)^n \quad (n = 1, 2, 3, \ldots)$.

The first 30 points on the graph of the sequence (y_n) can be plotted as follows.

First, define the sequence.

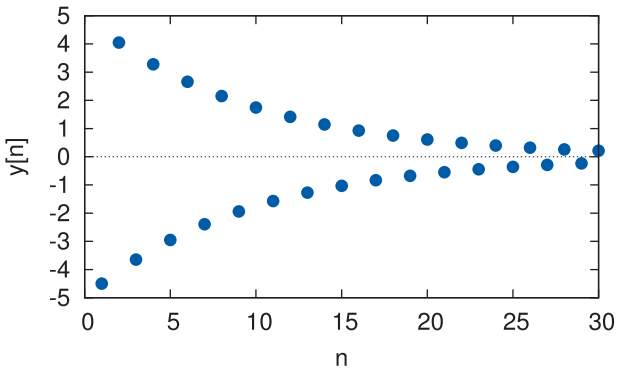
```
(%i4) y[n]:=5*(-0.9)^n;
```

(%o4) $y_n := 5(-0.9)^n$

Then generate the points to be plotted and plot them.

```
(%i5) ypts:makelist([n,y[n]],n,1,30)$
(%i6) wxplot2d([discrete,ypts], [style,points],
       [xlabel,"n"], [ylabel,"y[n]"]);
```

(%t6)



(%o6)

The sequence (y_n) alternates in sign and $y_n \to 0$ as $n \to \infty$.

(c) $z_n = -\frac{1}{10}(-2.5)^n \quad (n = 1, 2, 3, \ldots)$.

The first 15 points on the graph of the sequence (z_n) can be plotted as follows.

First, define the sequence.

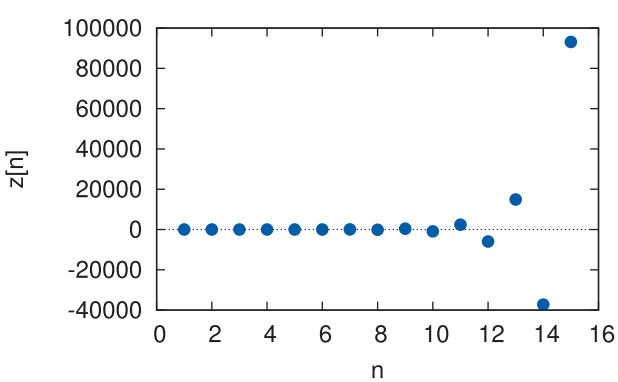
```
(%i7) z[n]:=-(1/10)*(-2.5)^n;
```

(%o7) $z_n := \left(-\frac{1}{10}\right)(-2.5)^n$

Then generate the points to be plotted and plot them.

```
(%i8) zpts:makelist([n,z[n]],n,1,15)$
(%i9) wxplot2d([discrete,zpts], [style,points],
       [xlabel,"n"], [ylabel,"z[n]"]);
```

(%t9)



(%o9)

The sequence (z_n) alternates in sign and is unbounded.

Computer solution to Unit 10, Activity 21

(a) $a_n = 17 - \frac{1}{3}(1.2)^n \quad (n = 1, 2, 3, \ldots)$.

The first 30 points on the graph of the sequence (a_n) can be plotted as follows.

First, define the sequence.

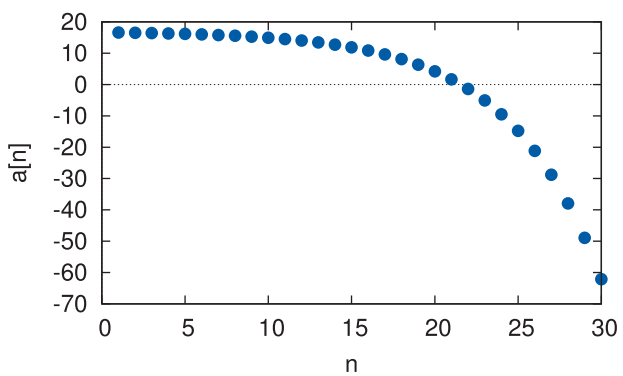
```
(%i1) a[n]:=17-(1/3)*(1.2)^n;
```

(%o1) $\mathtt{a_n} := 17 - \frac{1}{3}1.2^{\mathtt{n}}$

Then generate the points to be plotted and plot them.

```
(%i2) apts:makelist([n,a[n]],n,1,30)$
(%i3) wxplot2d([discrete,apts], [style,points],
       [xlabel,"n"], [ylabel,"a[n]"]);
```

(%t3)



(%o3)

The sequence (a_n) is decreasing and $a_n \to -\infty$ as $n \to \infty$.

(b) $b_n = 5(-0.9)^n + 45 \quad (n = 1, 2, 3, \ldots)$.

The first 30 points on the graph of the sequence (b_n) can be plotted as follows.

First, define the sequence.

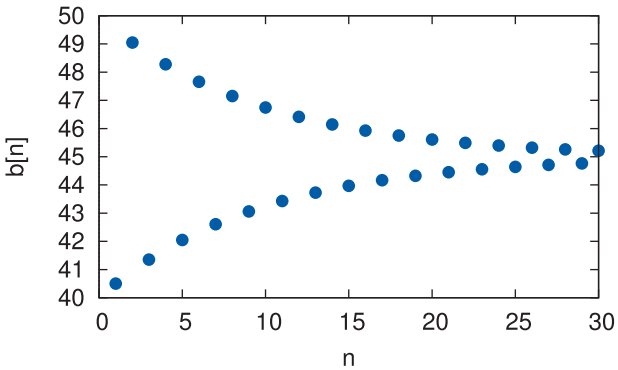
```
(%i4) b[n]:=5*(-0.9)^n+45;
```

(%o4) $\mathtt{b_n} := 5(-0.9)^{\mathtt{n}} + 45$

Then generate the points to be plotted and plot them.

```
(%i5) bpts:makelist([n,b[n]],n,1,30)$
(%i6) wxplot2d([discrete,bpts], [style,points],
       [xlabel,"n"], [ylabel,"b[n]"]);
```

(%t6)



(%o6)

The terms of (b_n) alternate either side of 45 and $b_n \to 45$ as $n \to \infty$.

Solutions to Computer activities

Solution to Computer activity 3

```
(%i1) 2+3*4/5;
```

(%o1) $\frac{22}{5}$

So $2 + 3 \times \frac{4}{5} = \frac{22}{5}$.

Solution to Computer activity 5

(a)
```
(%i1) 27^(1/3);
(%o1) 3
```

(b)
```
(%i2) 27^1/3;
(%o2) 9
```

Without any brackets, `27^1/3` is interpreted according to the BIDMAS rules, as $(27^1)/3$ which is equal to 9.

Do not be surprised if your line numbering differs from ours. Each of our solutions is prepared as a separate Maxima session with the line numbering starting at 1 (unless it is a continuation of a previous activity); your numbering will continue to increase for the duration of your session.

Solution to Computer activity 7

(a)
```
(%i1) 1309/3451;
```

(%o1) $\frac{11}{29}$

(b)
```
(%i2) 1309.0/3451;
(%o2) 0.3793103448275862
```

So $1309.0/3451 \approx 0.379\,310\,344\,827\,5862$.

(c)
```
(%i3) float(1309/3451);
(%o3) 0.3793103448275862
```

So $1309/3451 \approx 0.379\,310\,344\,827\,5862$.

Your answers may differ in the last decimal place or so to those given here. This is due to small differences in the way floating-point numbers are computed by different systems.

Solution to Computer activity 8

(a)
```
(%i1) sqrt(147);
```
$$(\%o1)\quad 7\sqrt{3}$$
```
(%i2) float(sqrt(147));
```
```
(%o2) 12.12435565298214
```
So $\sqrt{147} = 7\sqrt{3} \approx 12.124\,355\,652\,982\,14$.

(b)
```
(%i3) (8/5)^(2/3);
```
$$(\%o3)\quad \frac{4}{5^{2/3}}$$
```
(%i4) float((8/5)^(2/3));
```
```
(%o4) 1.367980757341358
```
So $\left(\frac{8}{5}\right)^{2/3} = \frac{4}{5^{2/3}} \approx 1.367\,980\,757\,341\,358$.

(c)
```
(%i5) 2^500;
```
```
(%o5) 327339060789614187001318969682[91digits]
      217256545885393053328527589376
```
```
(%i6) float(2^500);
```
$$(\%o6)\quad 3.273390607896142\ \ 10^{150}$$
So $2^{500} \approx 3.273\,390\,607\,896\,142 \times 10^{150}$.

Solution to Computer activity 9

(a)
```
(%i1) %pi+%pi/3+%e^2;
```
$$(\%o1)\quad \frac{4\pi}{3} + \%e^2$$
```
(%i2) float(%pi+%pi/3+%e^2);
```
```
(%o2) 11.57784630371704
```
So $\pi + \frac{\pi}{3} + e^2 = \frac{4\pi}{3} + e^2 \approx 11.577\,846\,303\,717\,04$.

Solution to Computer activity 10

(a)
```
(%i1) 2*sqrt(3)+5*sqrt(27);
```
$$(\%o1)\quad 17\sqrt{3}$$
So $2\sqrt{3} + 5\sqrt{27} = 17\sqrt{3}$.

(b)
```
(%i2) 2*sqrt(3)-5*sqrt(27);
```
$$(\%o2)\quad -13\sqrt{3}$$
So $2\sqrt{3} - 5\sqrt{27} = -13\sqrt{3}$.

Solution to Computer activity 11

At the end of this activity, your worksheet should look like the following, although line numbers may differ.

```
(%i4) 6*2;
(%o4) 12
(%i5) 3+%;
(%o5) 15
(%i6) %o4+5;
(%o6) 17
```

Solution to Computer activity 12

(a)
```
(%i1) a:sqrt(8);
```
(%o1) $2^{3/2}$

(b)
```
(%i2) a;
```
(%o2) $2^{3/2}$

(c)
```
(%i3) b:a*sqrt(2);
(%o3) 4
```

(d)
```
(%i4) a:sqrt(7);
```
(%o4) $\sqrt{7}$

(e)
```
(%i5) b;
(%o5) 4
```

(f)
```
(%i6) values;
```
(%o6) [a, b]

(g)
```
(%i7) kill(a);
(%o7) done
```

(h)
```
(%i8) a;
(%o8) a
```

(i)
```
(%i9) b;
(%o9) 4
```

Solution to Computer activity 13

(a)
```
(%i1) a:42;
(%o1) 42
```

(b)
```
(%i2) b:a^3-3*a+4;
(%o2) 73966
```

(c)
```
(%i3) c:float(sqrt(b));
(%o3) 271.9669097519035
```

Solution to Computer activity 14

(a)
```
(%i1) fpprintprec:8;
(%o1) 8
```

(This sets the number of significant figures to be shown in the display of decimal numbers.)

(b)
```
(%i2) float(%pi);
(%o2) 3.1415927
```

So the value of π is 3.141 5927 (to 8 s.f.).

Solution to Computer activity 15

The circumference of a circle of radius 5 is 31.42 (to 2 d.p.) and the area is 78.54 (to 2 d.p.).

Solution to Computer activity 17

(a)
```
(%i1) ? float;
-- Function: float (<expr>)
     Converts integers, rational numbers and bigfloats in
     <expr> to floating point numbers.  It is also an
     'evflag', 'float' causes non-integral rational numbers
     and bigfloat numbers to be converted to floating point.
  There are also some inexact matches for 'float'.
  Try '?? float' to see them.
(%o1) true
```

(b)
```
(%i2) ?? sqr;
 0: isqrt  (Functions and Variables for Number Theory)
 1: sqrt  (Root Exponential and Logarithmic Functions)
 2: sqrtdenest  (Package sqdnst)
 3: sqrtdispflag  (Functions and Variables for Display)
--> 1;
-- Function: sqrt (<x>)
     The square root of <x>.  It is represented internally by
     '<x>^(1/2)'.  See also 'rootscontract' and 'radexpand'.
(%o2) true
```

Solution to Computer activity 18

(a)
```
(%i1) expand( (2+x)*(3*x+4)^3 );
```
(%o1) $27x^4 + 162x^3 + 360x^2 + 352x + 128$

So $(2+x)(3x+4)^3 = 27x^4 + 162x^3 + 360x^2 + 352x + 128$.

(b)
```
(%i2) factor( 2*x^3-7*x^2-10*x+24 );
```
(%o2) $(x-4)(x+2)(2x-3)$

So $2x^3 - 7x^2 - 10x + 24 = (x-4)(x+2)(2x-3)$.

(c)
```
(%i3) p:(x^3+1)/(x+1);
```
(%o3) $\dfrac{x^3+1}{x+1}$

(d)
```
(%i4) fullratsimp(p);
```
(%o4) $x^2 - x + 1$

(e)
```
(%i5) subst(3,x,p);
```
(%o5) 7

Solution to Computer activity 19

(a)
```
(%i1) r:x^2+2*x=4;
```
(%o1) $x^2 + 2x = 4$

(b)
```
(%i2) lhs(r);
```
(%o2) $x^2 + 2x$

(c)
```
(%i3) rhs(r);
```
(%o3) 4

Solution to Computer activity 20

(a)
```
(%i1) solve( x^2+2*x=1 );
```
(%o1) $[x = -\sqrt{2} - 1, x = \sqrt{2} - 1]$

The solutions of the equation are $x = -\sqrt{2} - 1$ and $x = \sqrt{2} - 1$.

(b)
```
(%i2) solve( 4*x^2-12*x+9=0 );
```
(%o2) $[x = \dfrac{3}{2}]$

The solution of the equation is $x = \frac{3}{2}$.

(c)
```
(%i3) solve( 2*x^3+x^2-5*x+2=0 );
```
(%o3) $[x = \dfrac{1}{2}, x = -2, x = 1]$

The solutions of the equation are $x = \frac{1}{2}$, $x = -2$ and $x = 1$.

(d) `(%i4) solve( x^9+2*x-4=0 );`

`(%o4)` $[0 = x^9 + 2x - 4]$

Maxima was unable to solve this equation.

(e) `(%i5) solve( x^2-4*x+5=0 );`

`(%o5)` $[x = 2 - \%i, x = \%i + 2]$

The two complex solutions are $x = 2 - i$ and $x = 2 + i$. There are no real solutions.

Solution to Computer activity 21

(a) `(%i1) A: [2,3,5,7,11];`

`(%o1)` $[2, 3, 5, 7, 11]$

(b) `(%i2) A[4];`

`(%o2)` 7

So the fourth element of `A` is 4.

(c) (i) `(%i3) solns:solve(x^2-x-1=0);`

`(%o3)` $[x = -\frac{\sqrt{5}-1}{2}, x = \frac{\sqrt{5}+1}{2}]$

So the solutions are $x = -\frac{\sqrt{5}-1}{2}$ and $x = \frac{\sqrt{5}+1}{2}$.

(ii) `(%i4) float(solns);`

`(%o4)` $[x = -0.61803398874989, x = 1.618033988749895]$

So decimal approximations to the solutions are $x = -0.618\,033\,988\,749\,89$ and $x = 1.618\,033\,988\,749\,895$.

(iii) `(%i5) rhs(solns[1]);`

`(%o5)` $-\frac{\sqrt{5}-1}{2}$

Solution to Computer activity 22

(a) `(%i1) eq:3*y+4=3*x^2+9*x;`

`(%o1)` $3y + 4 = 3x^2 + 9x;$

(b) `(%i2) eq-4;`

`(%o2)` $3y = 3x^2 + 9x - 4;$

(c) `(%i3) %/3;`

`(%o3)` $y = \frac{3x^2 + 9x - 4}{3}$

Solution to Computer activity 23

(a)
```
(%i1) eqn:c=(2*a+b+3*c)/(a-b);
```
(%o1) $c = \dfrac{3c + b + 2a}{a - b}$

(b)
```
(%i2) solve(eqn, a);
```
(%o2) $[a = \dfrac{(b + 3)c + b}{c - 2}]$

So $a = \dfrac{(b+3)c+b}{c-2}$.

Solution to Computer activity 24

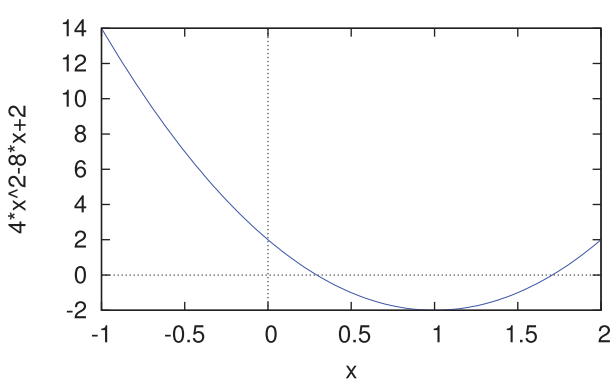
(a)
```
(%i1) eq1:3*x^2+2*y=20;
```
(%o1) $2y + 3x^2 = 20$
```
(%i2) eq2:4*x-3*y=-4;
```
(%o2) $4x - 3y = -4$

(b)
```
(%i3) solve( [eq1, eq2], [x,y] );
```
(%o3) $[[x = 2, y = 4], [x = -\dfrac{26}{9}, y = -\dfrac{68}{27}]]$

The two solutions are $x = 2,\ y = 4$ and $x = -\frac{26}{9},\ y = -\frac{68}{27}$.

Solution to Computer activity 25

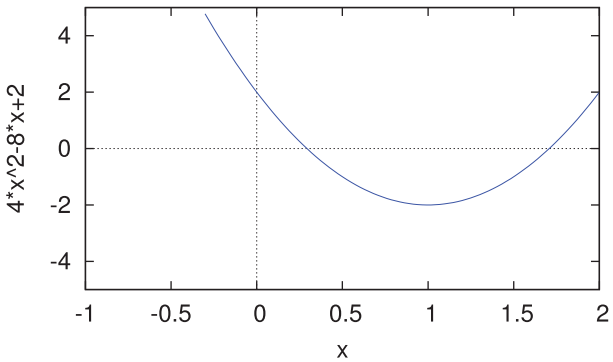
(a)
```
(%i1) wxplot2d(4*x^2-8*x+2, [x,-1,2]);
```
(%t1)



(%o1)

(b)
```
(%i2) wxplot2d(4*x^2-8*x+2, [x,-1,2], [y,-5,5]);
plot2d: some values were clipped.
```

(%t2)

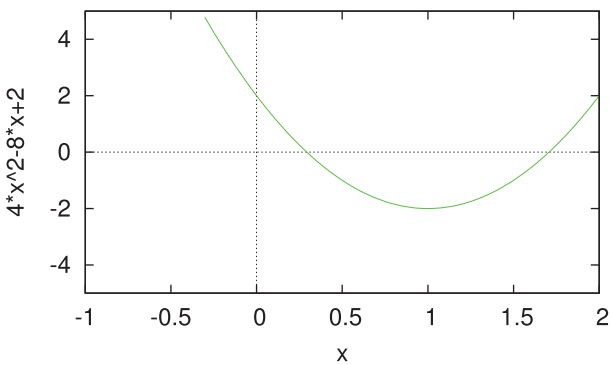


(%o2)

(c)
```
(%i3) wxplot2d(4*x^2-8*x+2, [x,-1,2], [y,-5,5],
       [color, green]);
```

```
plot2d: some values were clipped.
```

(%t3)

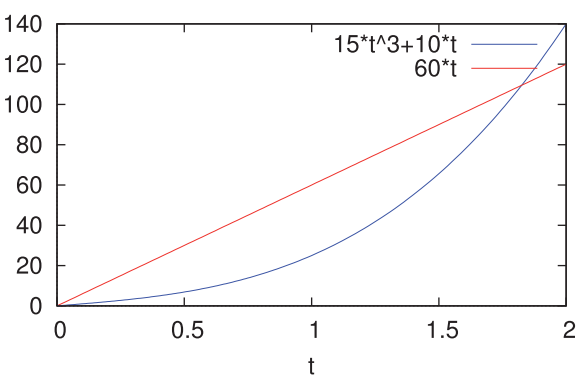


(%o3)

Solution to Computer activity 26

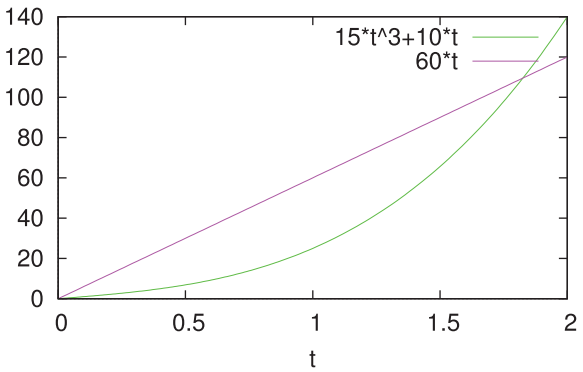
(a)
```
(%i1) wxplot2d([15*t^3+10*t, 60*t], [t,0,2]);
```

(%t1)



(%o1)

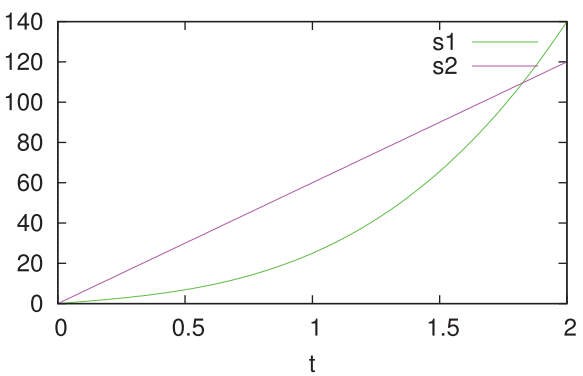
(b)
```
(%i2) wxplot2d([15*t^3+10*t, 60*t], [t,0,2],
       [color, green, magenta]);
```



(%t2)

(%o2)

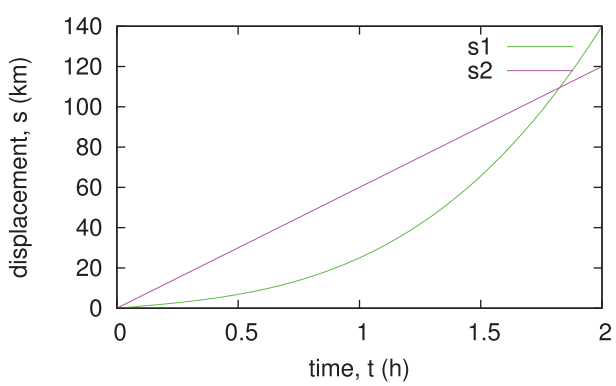
(c)
```
(%i3) wxplot2d([15*t^3+10*t, 60*t], [t,0,2],
       [color, green, magenta],
       [legend, "s1", "s2"]);
```



(%t3)

(%o3)

(d)
```
(%i4) wxplot2d([15*t^3+10*t, 60*t], [t,0,2],
       [color, green, magenta],
       [legend, "s1", "s2"],
       [xlabel, "time, t (h)"],
       [ylabel, "displacement, s (km)"]);
```



(%t4)

(%o4)

Solution to Computer activity 27

(a)
```
(%i1) f(x):=x^5-3*x^2+4;
```
(%o1) $f(x) := x^5 - 3x^2 + 4;$

(b)
```
(%i2) f(2);
```
(%o2) 24

So $f(2) = 24$.

(c)
```
(%i3) g(x):= (x-1)/(x-2);
```
(%o3) $g(x) := \dfrac{x-1}{x-2};$
```
(%i4) g(3);
```
(%o4) 2

So $g(3) = 2$.

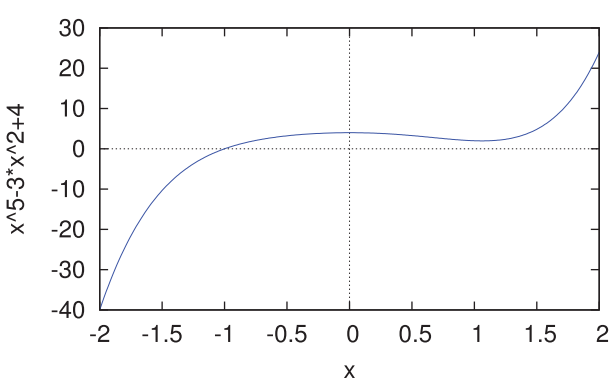
(d)
```
(%i5) g(2);
expt: undefined: 0 to a negative exponent.
#0: g(x=2)
-- an error. To debug this try: debugmode(true);
```

(e)
```
(%i6) h(1);
```
(%o6) h(1);

(f)
```
(%i7) functions;
```
(%o7) [f(x), g(x)]

Solution to Computer activity 28

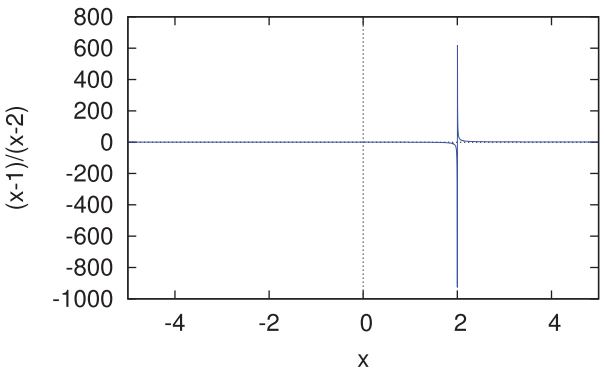
(a)
```
(%i8) wxplot2d(f(x), [x,-2,2]);
```
(%t8)



(%o8)

(b)
```
(%i9) wxplot2d(g(x), [x,-5,5]);
```

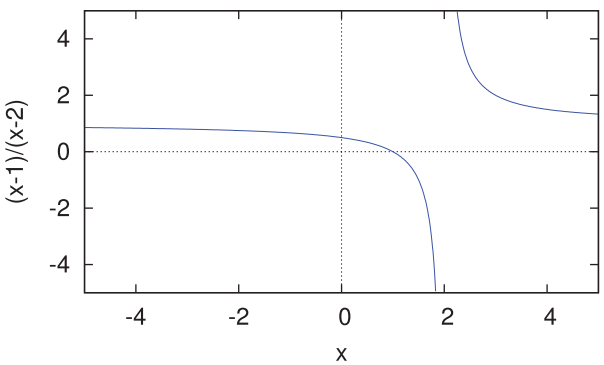
(%t9)



(%o9)

(c)
```
(%i10) wxplot2d(g(x), [x,-5,5], [y,-5,5]);
plot2d:  some values were clipped.
```

(%t10)



(%o10)

Solution to Computer activity 29

(a)
```
(%i1) %e^5;
```
(%o1) $\%e^5$;

(b)
```
(%i2) float(%e^5);
```
(%o2) 148.4131591025766

So $e^5 = 148.41$ (to 5 s.f.).

(c)
```
(%i3) exp(5);
```
(%o3) $\%e^5$;

(d)
```
(%i4) float( log(8) );
```
(%o4) 2.079441541679836

So $\log(8) = 2.079$ (to 4 s.f.).

(e)
```
(%i5) solve( 3^(2*x)=5 )
```

(%o5) $[\mathtt{x} = \frac{\log(-\sqrt{5})}{\log(3)}, \mathtt{x} = \frac{\log(5)}{2\log(3)}]$

The only real solution is $\frac{\log(5)}{2\log(3)}$.

Solution to Computer activity 30

(a)
```
(%i1) u:( log(x^3+2*x^2+x)-log(x) )/log(x+1);
```
(%o1) $\dfrac{\log(x^3+2x^2+x)-\log(x)}{\log(x+1)}$

(b)
```
(%i2) radcan(u);
```
(%o2) 2

Notice,

$$\begin{aligned}\frac{\ln(x^3+2x^2+x)-\ln(x)}{\ln(x+1)} &= \frac{1}{\ln(x+1)}\left(\ln(x^3+2x^2+x)-\ln(x)\right)\\ &= \frac{1}{\ln(x+1)}\left(\ln\left(\frac{x^3+2x^2+x}{x}\right)\right)\\ &= \frac{1}{\ln(x+1)}\left(\ln(x^2+2x+1)\right)\\ &= \frac{1}{\ln(x+1)}\left(\ln(x+1)^2\right)\\ &= \frac{2\ln(x+1)}{\ln(x+1)}\\ &= 2.\end{aligned}$$

(c)
```
(%i3) v:log(a)+2*log(b)-log(c);
```
(%o3) $-\log(c)+2\log(b)+\log(a)$

(d)
```
(%i4) logcontract(v);
```
(%o4) $\log\left(\dfrac{ab^2}{c}\right)$

Solution to Computer activity 31

(a)
```
(%i1) cos(%pi/6);
```
(%o1) $\dfrac{\sqrt{3}}{2}$

So $\cos\left(\dfrac{\pi}{6}\right)=\dfrac{\sqrt{3}}{2}$.

(b)
```
(%i2) sin(45*%pi/180);
```
(%o2) $\dfrac{1}{\sqrt{2}}$

So $\sin 45^\circ = \dfrac{1}{\sqrt{2}}$.

(c)
```
(%i3) sec(3.4);
```
(%o3) −1.034342024711439

So $\sec(3.4) \approx -1.034\,342\,024\,711\,439$.

(d) `(%i4) asin(sqrt(3)/2);`

`(%o4)` $\frac{\pi}{3}$

So $\sin^{-1}\left(\frac{\sqrt{3}}{2}\right) = \frac{\pi}{3}$.

(e) `(%i5) csc(%pi/7);`

`(%o5)` $\csc\left(\frac{\pi}{7}\right)$

(f) `(%i6) tan(-%pi/12);`

`(%o6)` $-\tan\left(\frac{\pi}{12}\right)$

(g) `(%i7) acos(2.0);`

`(%o7) 1.316957896924817%i`

So $\cos^{-1} 2.0 \approx 1.316\,957\,896\,924\,817i$.

(h) `(%i8) acos(2);`

`(%o8) acos(2);`

Solution to Computer activity 32

(a) `(%i1) solve(cos(x)-0.6=0);`

```
rat: replaced -0.6 by -3/5 = -0.6
solve: using arc-trig functions to get a solution.
Some solutions will be lost.
```

`(%o1)` $[\mathrm{x} = \mathrm{acos}\left(\frac{3}{5}\right)]$

So a solution is $x = \cos^{-1}\left(\frac{3}{5}\right)$.

(b) `(%i2) float(%)`

`(%o2)` $[\mathrm{x} = 0.9272952180016121]$

So a solution is $x = 0.927$ (to 3 s.f.).

(c) `(%i3) solve(sin(3*x+4)=1/2);`

```
solve: using arc-trig functions to get a solution.
Some solutions will be lost.
```

`(%o3)` $[\mathrm{x} = \frac{\pi - 24}{18}]$

So a solution is $x = \frac{\pi - 24}{18}$.

(d) `(%i4) solve(cos(2*x+4)-sin(x/3)=0);`

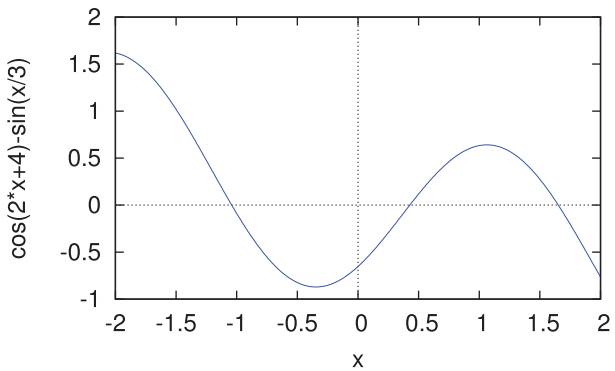
`(%o4)` $[\sin\left(\frac{\mathrm{x}}{3}\right) = \cos(2\mathrm{x} + 4)]$

Maxima cannot find an exact solution of this equation, so returns the equation slightly rearranged.

Solution to Computer activity 33

(a) (i)
```
(%i1) f(x):=cos(2*x+4)-sin(x/3);
```
(%o1) $f(x) := \cos(2x+4) - \sin\left(\frac{x}{3}\right)$

(ii)
```
(%i2) wxplot2d(f(x), [x,-2,2]);
```
(%t2)



(%o2)

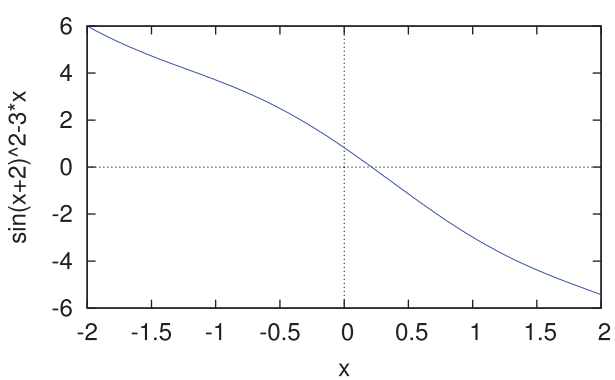
Since `f(0)` < 0 and `f(1)` > 0, there must be a solution in the interval $(0, 1)$.

(iii)
```
(%i3) find_root(f(x), x, 0, 1);
```
(%o3) 0.4274333882308137

So a solution of $\cos(2x + 4) - \sin(x/3)$ within $(0, 1)$ is 0.427 (to 3 s.f.).

(b)
```
(%i4) g(x):= (sin(x+2))^2-3*x;
```
(%o4) $g(x) := \sin(x+2)^2 - 3x$

```
(%i5) wxplot2d(g(x), [x,-2,2]);
```
(%t5)



(%o5)

Since `g(0)` > 0 and `g(1)` < 0, there must be a solution in the interval $(0, 1)$.
(If you chose an interval other than $(-2, 2)$ then your graph will look different from the one given above. If, in particular, your graph does not cross the x-axis, then you will need to plot the graph of g again using a larger interval.)

```
(%i6) find_root(g(x), x, 0, 1);
```

```
(%o6)  0.2135670173061208
```

So a solution of $\sin^2(x+2) = 3x$ is $x = 0.214$ (to 3 s.f.).

Solution to Computer activity 34

(a)
```
(%i1)  trigreduce( sin(x)^2 );
```
$$(\%o1)\quad \frac{1-\cos(2x)}{2}$$

(b)
```
(%i2)  trigexpand(%);
```
$$(\%o2)\quad \frac{\sin(x)^2-\cos(x)^2+1}{2}$$

(c)
```
(%i3)  trigsimp(%);
(%o3)  sin(x)^2
```

Solution to Computer activity 35

```
(%i1)  trigrat( sin(4*x)/sin(x) );
(%o1)  2cos(3x)+2cos(x)
```

So $\dfrac{\sin(4x)}{\sin(x)} = 2\cos(3x) + 2\cos(x)$.

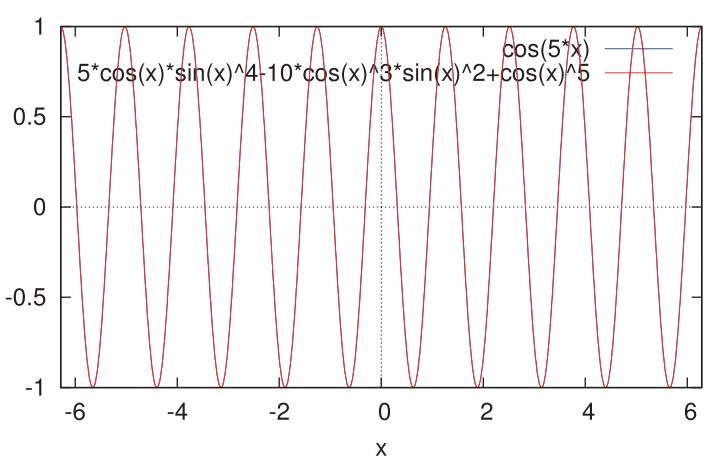
Solution to Computer activity 36

(a)
```
(%i1)  f:cos(5*x);
(%o1)  cos(5x)
(%i2)  g:trigexpand(f);
(%o2)  5cos(x)sin(x)^4-10cos(x)^3sin(x)^2+cos(x)^5
```

So $\cos(5x) = 5\cos(x)\sin^4(x) - 10\cos^3(x)\sin^2(x) + \cos^5(x)$.

(b) Choosing the range $-2\pi \le x \le 2\pi$ to plot the graphs, we obtain

```
(%i3)  wxplot2d([f,g],[x, -2*%pi, 2*%pi]);
```



(%t3)

(%o3)

The two graphs appear identical, which suggests the two expressions are equal.

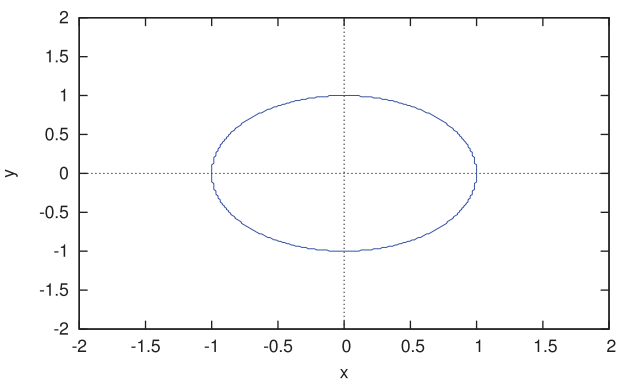
Solution to Computer activity 37

(a)
```
(%i1) load(implicit_plot);
(%o1) C:\maxima-5.38.1\share\maxima\5.38.1_5\share\contrib\
      implicit_plot.lisp
```

(The output from this command is the location of the package on your computer system, and may differ from that shown here.)

(b)
```
(%i2) wximplicit_plot(x^2+y^2=1, [x,-2,2], [y,-2,2]);
```

(%t2)



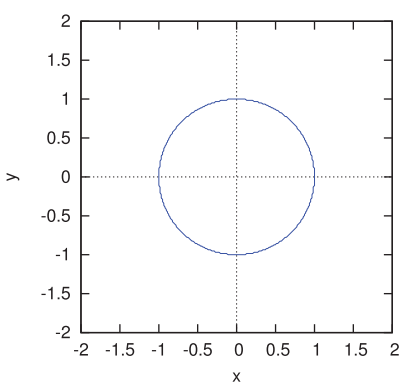
(%o2)

Solution to Computer activity 38

(It is assumed the `implicit_plot` package has already been loaded.)

```
(%i1) wximplicit_plot(x^2+y^2=1, [x,-2,2], [y,-2,2], same_xy);
```

(%t1)



(%o1)

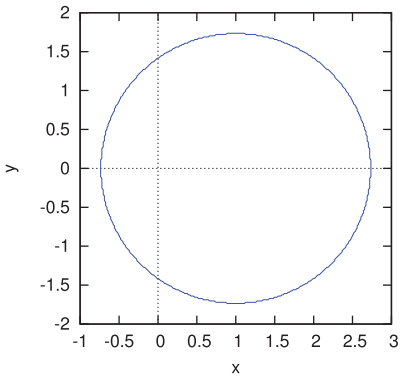
Solution to Computer activity 39

(It is assumed the `implicit_plot` package has already been loaded.)

(a)
```
(%i1) set_plot_option(same_xy)$
```

(b)
```
(%i2) wximplicit_plot((x-1)^2+y^2=3, [x,-1,3], [y,-2,2]);
```

(%t2)



(%o2)

Solution to Computer activity 40

(It is assumed the `implicit_plot` package has already been loaded, and the plot settings changed to use equal scales for the x- and y-axes.)

(a) `(%i1) line:y=3*x-5;`

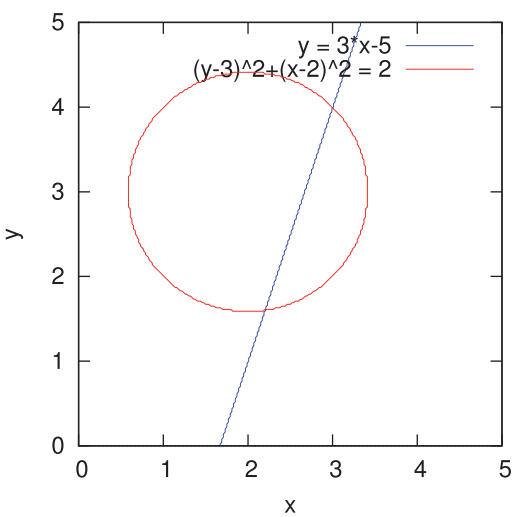
`(%o1)` $y = 3x - 5$

`(%i2) circle:(x-2)^2+(y-3)^2=2;`

`(%o2)` $(y-3)^2 + (x-2)^2 = 2$

(b) `(%i3) wximplicit_plot([line, circle], [x,0,5], [y,0,5]);`

(%t3)



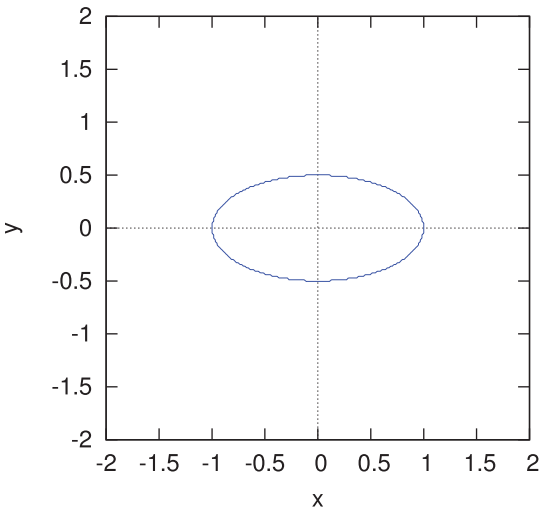
(%o3)

Solution to Computer activity 41

(It is assumed the `implicit_plot` package has previously been loaded and the plotting options set for equal-scale axes.)

(a) `(%i1) wximplicit_plot(x^2+4*y^2=1, [x,-2,2], [y,-2,2]);`

(%t1)

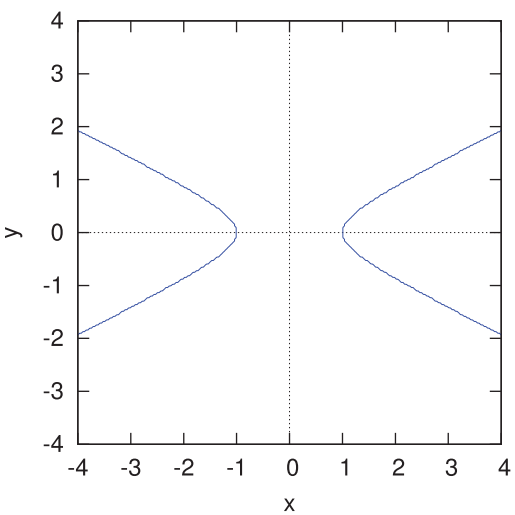


(%o1)

(The curve is an *ellipse*.)

(b) `(%i2) wximplicit_plot(x^2-4*y^2=1, [x,-4,4], [y,-4,4]);`

(%t2)

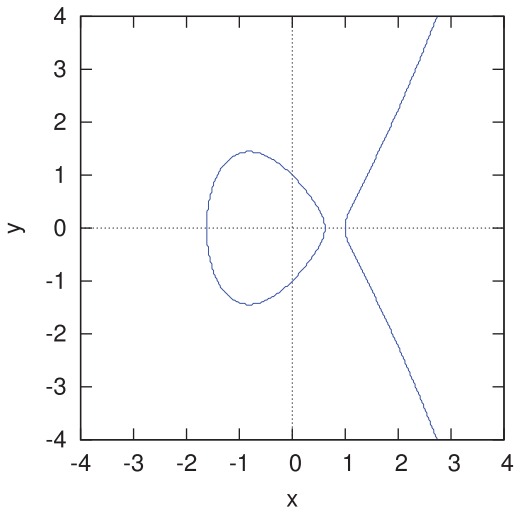


(%o2)

(The curve is a *hyperbola*.)

(c) `(%i3) wximplicit_plot(y^2=x^3-2*x+1, [x,-4,4], [y,-4,4]);`

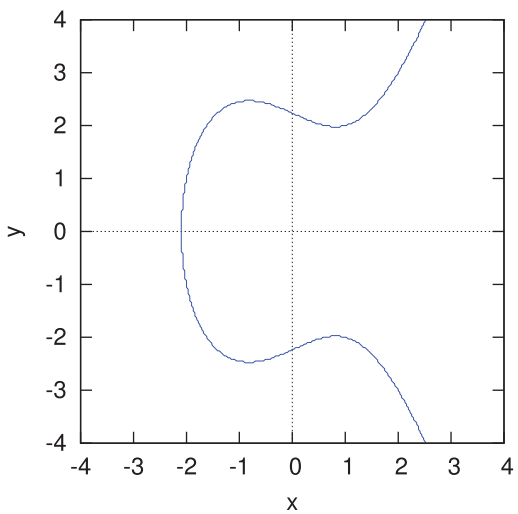
(%t3)



(%o3)

(d) (%i4) `wximplicit_plot(y^2=x^3-2*x+5, [x,-4,4], [y,-4,4]);`

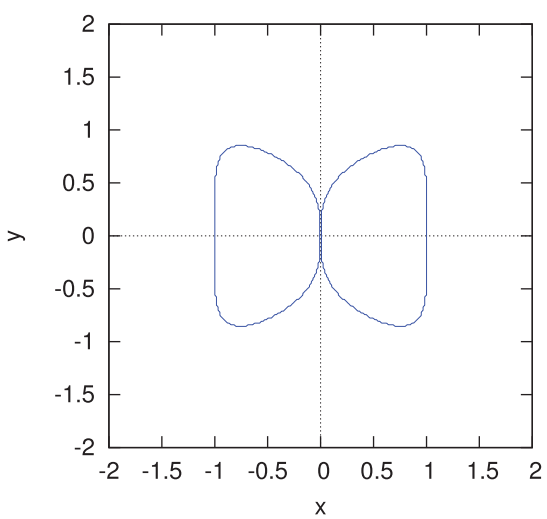
(%t4)



(%o4)

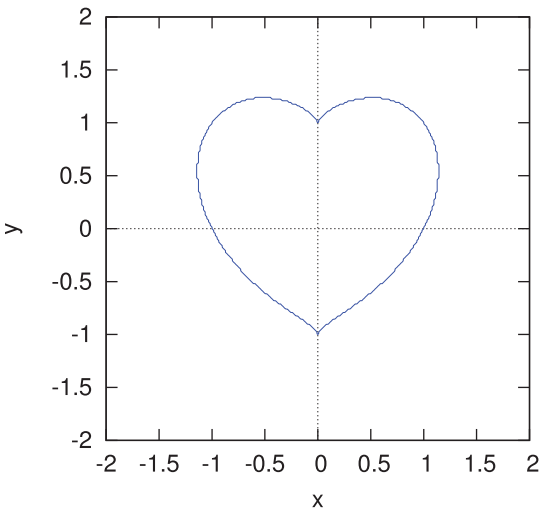
(e) (%i5) `wximplicit_plot(y^6=x^2-x^6, [x,-2,2], [y,-2,2]);`

(%t5)



(%o5)

(f)
```
(%i6) wximplicit_plot((x^2+y^2-1)^3=x^2*y^3, [x,-2,2], [y,-2,2]);
```
(%t6)



(%o6)

Solution to Computer activity 42

(a)
```
(%i1) diff(x^4+5*x^2+15, x);
```
(%o1) $4\,x^3+10\,x$

So the derivative of x^4+5x^2+15 is $4x^3+10x$.

(b)
```
(%i2) p:t^2/sin(4*t);
```
(%o2) $\dfrac{t^2}{\sin(4\,t)}$
```
(%i3) diff(p,t);
```
(%o3) $\dfrac{2\,t}{\sin(4\,t)}-\dfrac{4\,t^2\cos(4\,t)}{\sin(4\,t)^2}$

So the derivative of $\dfrac{t^2}{\sin(4t)}$ is $\dfrac{2t}{\sin(4t)}-\dfrac{4t^2\cos(4t)}{\sin^2(4t)}$.

(c)
```
(%i4) g(u):=(u^2+3)*log(u^2);
```
(%o4) $g(u):=(u^2+3)\log(u^2)$
```
(%i5) diff(g(u),u);
```
(%o5) $4\,u\log(u)+\dfrac{2(u^2+3)}{u}$

So the derivative of $(u^2+3)\ln(u^2)$ is $4u\ln(u)+\dfrac{2(u^2+3)}{u}$.

(d)
```
(%i6) diff(p,t,2);
```
(%o6) $\dfrac{16\,t^2}{\sin(4\,t)}+\dfrac{2}{\sin(4\,t)}-\dfrac{16\,t\cos(4\,t)}{\sin(4\,t)^2}+\dfrac{32\,t^2\cos(4\,t)^2}{\sin(4\,t)^3}$

So the second derivative of $\dfrac{t^2}{\sin(4t)}$ is

$$\frac{16t^2}{\sin(4t)} + \frac{2}{\sin(4t)} - \frac{16t\cos(4t)}{\sin^2(4t)} + \frac{32t^2\cos^2(4t)}{\sin^3(4t)}.$$

(e)
```
(%i7) diff(g(u),u,3);
```
(%o7) $\dfrac{4\,(u^2+3)}{u^3}$

So the third derivative of $(u^2+3)\ln(u^2)$ is $\dfrac{4(u^2+3)}{u^3}$.

Solution to Computer activity 43

(a)
```
(%i1) diff(2*x^6+5/x, x);
```
(%o1) $12x^5 - \dfrac{5}{x^2}$

The derivative of $2x^6 + \dfrac{5}{x}$ is $12x^5 - \dfrac{5}{x^2}$.

(b)
```
(%i2) diff(log(exp(x)+x), x);
```
(%o2) $\dfrac{\%e^x + 1}{\%e^x + x}$

The derivative of $\ln(e^x + x)$ is $\dfrac{e^x+1}{e^x+x}$.

(c)
```
(%i3) diff((tan(t)-t^2)/exp(t), t);
```
(%o3) $\%e^{-t}(\sec(t)^2 - 2t) - \%e^{-t}(\tan(t) - t^2)$

This can be simplified as follows.

```
(%i4) fullratsimp(%);
```
(%o4) $-\%e^{-t}(\tan(t) - \sec(t)^2 - t^2 + 2t)$

(Remember, % in input line 4 refers to the result of the last evaluated command.)

The derivative of $\dfrac{\tan(t) - t^2}{e^t}$ is $-e^{-t}\left(\tan(t) - \sec(t)^2 - t^2 + 2t\right)$.

Solution to Computer activity 44

(a)
```
(%i1) f(x):=sin(x)/(exp(x)+1);
```
(%o1) $f(x) := \dfrac{\sin(x)}{\exp(x)+1}$

(b)
```
(%i2) df(x):=diff(f(x),x);
```
(%o2) $df(x) := \dfrac{d}{dx}f(x)$

(c)
```
(%i3) df(3);
diff:  second argument must be a variable; found 3
#0:  df(x=3)
-- an error.  To debug this try:  debugmode(true);
```

(d)
```
(%i4) df(x):=''(diff(f(x),x));
```
(%o4) $\texttt{df(x)} := \dfrac{\cos(\texttt{x})}{\%\texttt{e}^{\texttt{x}}+1} - \dfrac{\%\texttt{e}^{\texttt{x}}\sin(\texttt{x})}{(\%\texttt{e}^{\texttt{x}}+1)^2}$

So $f'(x) = \dfrac{\cos(x)}{e^x+1} - \dfrac{e^x\sin(x)}{(e^x+1)^2}$

(e)
```
(%i5) df(3);
```
(%o5) $\dfrac{\cos(3)}{\%\texttt{e}^3+1} - \dfrac{\%\texttt{e}^3\sin(3)}{(\%\texttt{e}^3+1)^2}$
```
(%i6) float(%);
```
(%o6) −0.05332658917585964

So the value of f' at $x = 3$ is -0.053 (to 2 s.f.).

(f) First, define the function.
```
(%i7) f(x):=sin(x)*exp(cos(x));
```
(%o7) f(x) := sin(x)exp(cos(x));

Then calculate the derivative, defining `df(x)` to be the result.
```
(%i8) df(x):=''(diff(f(x),x));
```
(%o8) $\texttt{df(x)} := \%\texttt{e}^{\cos(\texttt{x})}\cos(\texttt{x}) - \%\texttt{e}^{\cos(\texttt{x})}\sin(\texttt{x})^2$

Finally evaluate the derivative at $x = 1$, giving the result as a decimal approximation.
```
(%i9) float(df(1));
```
(%o9) −0.2879834260858308

So the derivative of $\sin(x)e^{\cos(x)}$ at $x = 1$ is -0.288 (to 3 s.f.).

Solution to Computer activity 45

(a)
```
(%i1) integrate(2*x^2+exp(3*x), x);
```
(%o1) $\dfrac{\%\texttt{e}^{3\texttt{x}}}{3} + \dfrac{2\,\texttt{x}^3}{3}$

So $\displaystyle\int 2x^2 + e^{3x}\,\mathrm{d}x = \frac{e^{3x}}{3} + \frac{2x^3}{3} + c.$

(b)
```
(%i2) integrate(1/x, x, 2, 3);
```
(%o2) log(3) − log(2)
```
(%i3) float(%);
```
(%o3) 0.4054651081081645

So $\displaystyle\int_2^3 \frac{1}{x}\,\mathrm{d}x = \ln(3) - \ln(2) = 0.405$ (to 3 s.f.).

(c)
```
(%i4) p(x):=x/(x^2+1);
```
(%o4) $\texttt{p(x)} := \dfrac{\texttt{x}}{\texttt{x}^2+1}$

```
(%i5) q(x):=''(integrate(p(x),x));
```

(%o5) $\mathtt{q(x)} := \dfrac{\mathtt{log(x^2+1)}}{2}$

So $\displaystyle\int p(x)\,\mathrm{d}x = \frac{\ln(x^2+1)}{2} + c.$

Solution to Computer activity 46

(a)
```
(%i1) integrate((x-3)*(x-1), x);
```

(%o1) $\dfrac{\mathtt{x^3 - 6\,x^2 + 9\,x}}{3}$

So $\displaystyle\int (x-3)(x-1)\,\mathrm{d}x = \frac{x^3 - 6x^2 + 9x}{3} + c.$

(b)
```
(%i2) integrate(1/sqrt(9-t^2), t);
```

(%o2) $\mathtt{asin}\left(\dfrac{\mathtt{t}}{3}\right)$

So $\displaystyle\int \frac{1}{\sqrt{9-t^2}}\,\mathrm{d}t = \sin^{-1}\left(\frac{t}{3}\right) + c.$

(c)
```
(%i3) integrate(exp(t)*sin(t), t, 0, %pi);
```

(%o3) $\dfrac{\%\mathtt{e}^{\pi}}{2} + \dfrac{1}{2}$

```
(%i4) float(%);
```

(%o4) 12.07034631638963

So $\displaystyle\int_0^{\pi} e^t \sin t\,\mathrm{d}t = \frac{e^{\pi}}{2} + \frac{1}{2} = 12.1$ (to 3 s.f.).

Solution to Computer activity 47

(a)
```
(%i1) integrate(exp(sin(x)), x);
```

(%o1) $\displaystyle\int \%\mathtt{e}^{\mathtt{sin(x)}}\mathtt{dx}$

Maxima was unable to calculate this integral.

(b)
```
(%i2) integrate(exp(-x^2), x);
```

(%o2) $\dfrac{\sqrt{\pi}\,\mathtt{erf(x)}}{2}$

Maxima has calculated this integral in terms of a function called the *Error function* and denoted `erf`.

(c)
```
(%i3) integrate(exp(x)/x, x);
```

(%o3) $-\mathtt{gamma_incomplete}(0, -\mathtt{x})$

Maxima has calculated this integral in terms of a function called the *incomplete Gamma function*.

Solution to Computer activity 48

(a) (i)
```
(%i1) kill(n);
(%o1) done
```

(ii)
```
(%i2) integrate(x^n, x);
Is n equal to -1?  n;
```
$$(\%o2)\quad \frac{x^{n+1}}{n+1}$$

(b) (i)
```
(%i3) assume(notequal(n,-1));
(%o3) [notequal(n, -1)]
```

(ii)
```
(%i4) integrate(x^n, x);
```
$$(\%o4)\quad \frac{x^{n+1}}{n+1}$$

(iii)
```
(%i5) facts();
(%o5) [notequal(n, -1)]
```
So Maxima knows $n \neq -1$.

(iv)
```
(%i6) forget(notequal(n,-1));
(%o6) [notequal(n, -1)]
```

(v)
```
(%i7) facts();
```
No facts are now known.

Solution to Computer activity 49

First, tell Maxima what we know about `a`.

```
(%i1) assume(a>0);
```
$$(\%o1)\quad [a > 0]$$

Then do the integration.

```
(%i2) integrate(1/(x^2+a),x);
```
$$(\%o2)\quad \frac{\operatorname{atan}\left(\frac{x}{\sqrt{a}}\right)}{\sqrt{a}}$$

So
$$\int \frac{1}{x^2+a}\,\mathrm{d}x = \frac{1}{\sqrt{a}}\tan^{-1}\left(\frac{x}{\sqrt{a}}\right) + c,$$
where $a > 0$.

Solution to Computer activity 50

(a)
```
(%i1) integrate(log(cos(x^2)), x, 0, 1);
```
$$(\%o1)\quad \int_0^1 \log(\cos(x^2))\,\mathrm{d}x$$

Maxima cannot evaluate this integral exactly. It returns the integral unevaluated.

(b)
```
(%i2) quad_qags(log(cos(x^2)), x, 0, 1);
```
(%o2) $[-0.1115121024361476, 3.331001582652842\,10^{-11}, 21, 0]$

(Remember `3.331001582652842 10^-11` means $3.331\,001\,582\,652\,842 \times 10^{-11}$.)

So $\int_0^1 \ln(\cos(x^2))\,\mathrm{d}x = -0.112$ (to 3 s.f.). (Note that due to the inherent differences in how different computer systems compute floating-point arithmetic, there may be differences in the final few decimal places between your result and that shown here.)

(c)
```
(%i3) quad_qags(exp(-t^2), t, 0, 1);
```
(%o3) $[0.7468241328124271, 8.291413475940727\,10^{-15}, 21, 0]$

So $\int_0^1 e^{-t^2}\,\mathrm{d}t = 0.747$ (to 3 s.f.).

Solution to Computer activity 51

(a)
```
(%i1) P:matrix( [2,-3,3], [-2,1,2], [1,-1,4] );
```
(%o1) $\begin{bmatrix} 2 & -3 & 3 \\ -2 & 1 & 2 \\ 1 & -1 & 4 \end{bmatrix}$

(b)
```
(%i2) Q:matrix( [5,-3], [-1,-2], [0,4] );
```
(%o2) $\begin{bmatrix} 5 & -3 \\ -1 & -2 \\ 0 & 4 \end{bmatrix}$

(c)
```
(%i3) R:matrix( [1,-2,-1], [4,7,0], [-5,1,3] );
```
(%o3) $\begin{bmatrix} 1 & -2 & -1 \\ 4 & 7 & 0 \\ -5 & 1 & 3 \end{bmatrix}$

Solution to Computer activity 52

(a)
```
(%i4) matrix_size(Q);
```
(%o4) $[3, 2]$

So the matrix `Q` has size 3×2.

(b)
```
(%i5) R[3,2];
```
(%o5) 1

The element in the third row and second column of `R` is 1.

(c)
```
(%i6) R[3];
```
(%o6) $[-5, 1, 3]$

(d)
```
(%i7) R[3,2]:0;
(%o7) 0
(%i8) R;
```
$$(\%o8)\quad \begin{bmatrix} 1 & -2 & -1 \\ 4 & 7 & 0 \\ -5 & 0 & 3 \end{bmatrix}$$

Solution to Computer activity 53

(a)
```
(%i1) P:matrix([2,0],[3,-1],[-4,6]);
```
$$(\%o1)\quad \begin{bmatrix} 2 & 0 \\ 3 & -1 \\ -4 & 6 \end{bmatrix}$$
```
(%i2) Q:matrix([0,3],[5,-4],[1,3]);
```
$$(\%o2)\quad \begin{bmatrix} 0 & 3 \\ 5 & -4 \\ 1 & 3 \end{bmatrix}$$
```
(%i3) R:matrix([1,3],[2,4]);
```
$$(\%o3)\quad \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$
```
(%i4) S:matrix([1],[2]);
```
$$(\%o4)\quad \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$
```
(%i5) T:matrix([3,4]);
(%o5) [3  4]
```

(b)
```
(%i6) P+Q;
```
$$(\%o6)\quad \begin{bmatrix} 2 & 3 \\ 8 & -5 \\ -3 & 9 \end{bmatrix}$$

So $\mathbf{P}+\mathbf{Q} = \begin{pmatrix} 2 & 3 \\ 8 & -5 \\ -3 & 9 \end{pmatrix}$.

(c)
```
(%i7) P-R;
fullmap:  arguments must have same formal structure.
 -- an error.  To debug this try:  debugmode(true);
```

$\mathbf{P}-\mathbf{R}$ cannot be calculated, since the matrices are of different size. An error is given.

(d) `(%i8) 2*P;`

`(%o8)` $\begin{bmatrix} 4 & 0 \\ 6 & -2 \\ -8 & 12 \end{bmatrix}$

So $2\mathbf{P} = \begin{pmatrix} 4 & 0 \\ 6 & -2 \\ -8 & 12 \end{pmatrix}$.

(e) `(%i9) P.R;`

`(%o9)` $\begin{bmatrix} 2 & 6 \\ 1 & 5 \\ 8 & 12 \end{bmatrix}$

So $\mathbf{PR} = \begin{pmatrix} 2 & 6 \\ 1 & 5 \\ 8 & 12 \end{pmatrix}$.

(f) `(%i10) P.Q;`

```
MULTIPLYMATRICES: attempt to multiply nonconformable
matrices.
 -- an error.  To debug this try:  debugmode(true);
```

PQ cannot be calculated, since the matrices are not of the appropriate sizes. An error is given.

(g) `(%i11) P.S;`

`(%o11)` $\begin{bmatrix} 2 \\ 1 \\ 8 \end{bmatrix}$

So $\mathbf{PS} = \begin{pmatrix} 2 \\ 1 \\ 8 \end{pmatrix}$.

(h) `(%i12) P.T;`

`(%o12)` $\begin{bmatrix} 6 \\ 5 \\ 12 \end{bmatrix}$

Maxima interprets the row vector **T** as a column vector and obtains $\begin{pmatrix} 6 \\ 5 \\ 12 \end{pmatrix}$.

(i) `(%i13) R^^2;`

`(%o13)` $\begin{bmatrix} 7 & 15 \\ 10 & 22 \end{bmatrix}$

So $\mathbf{R}^2 = \begin{pmatrix} 7 & 15 \\ 10 & 22 \end{pmatrix}$.

(j) `(%i14) R^2;`

$$\texttt{(\%o14)}\ \begin{bmatrix} 1 & 9 \\ 4 & 16 \end{bmatrix}$$

Maxima returns the matrix $\begin{pmatrix} 1 & 9 \\ 4 & 16 \end{pmatrix}$ whose elements are the squares of corresponding elements of $\mathbf{R}$.

(k) `(%i15) P*Q;`

$$\texttt{(\%o15)}\ \begin{bmatrix} 0 & 0 \\ 15 & 4 \\ -4 & 18 \end{bmatrix}$$

Maxima returns the matrix $\begin{pmatrix} 0 & 0 \\ 15 & 4 \\ -4 & 18 \end{pmatrix}$ whose elements are the products of corresponding elements of $\mathbf{P}$ and $\mathbf{Q}$.

Solution to Computer activity 54

(a) `(%i1) A:matrix([1,2,0],[-4,4,2],[-1,2,1]);`

$$\texttt{(\%o1)}\ \begin{bmatrix} 1 & 2 & 0 \\ -4 & 4 & 2 \\ -1 & 2 & 1 \end{bmatrix}$$

`(%i2) B:matrix([1,2,0],[5,6,2],[2,2,1]);`

$$\texttt{(\%o2)}\ \begin{bmatrix} 1 & 2 & 0 \\ 5 & 6 & 2 \\ 2 & 2 & 1 \end{bmatrix}$$

(b) `(%i3) determinant(A);`

`(%o3) 4`

So $\det \mathbf{A} = 4$.

(c) `(%i4) C:invert(A);`

$$\texttt{(\%o4)}\ \begin{bmatrix} 0 & -\frac{1}{2} & 1 \\ \frac{1}{2} & \frac{1}{4} & -\frac{1}{2} \\ -1 & -1 & 3 \end{bmatrix}$$

So $\mathbf{A}^{-1} = \begin{pmatrix} 0 & -\frac{1}{2} & 1 \\ \frac{1}{2} & \frac{1}{4} & -\frac{1}{2} \\ -1 & -1 & 3 \end{pmatrix}$.

(d) `(%i5) A.C;`

$$\texttt{(\%o5)}\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
(%i6) C.A;
```

$$(\%o6)\quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

So $\mathbf{AC} = \mathbf{CA} = \mathbf{I}$, and $\mathbf{C}$ is the inverse of $\mathbf{A}$.

(e)
```
(%i7) invert(B);
expt:  undefined:  0 to a negative exponent.
 -- an error.  To debug this try:  debugmode(true);
```

$\mathbf{B}$ does not have an inverse.

Solution to Computer activity 55

(a) First, enter the matrix into Maxima, and assign it to a variable.

```
(%i1) A:matrix([2,-1,1],[2,0,1],[4,2,1]);
```

$$(\%o1)\quad \begin{bmatrix} 2 & -1 & 1 \\ 2 & 0 & 1 \\ 4 & 2 & 1 \end{bmatrix}$$

Then, check if the matrix is invertible by finding its determinant.

```
(%i2) determinant(A);
```

$$(\%o2)\quad -2$$

We have

$$\det \begin{pmatrix} 2 & -1 & 1 \\ 2 & 0 & 1 \\ 4 & 2 & 1 \end{pmatrix} = -2,$$

so this matrix is invertible.

Now find the inverse.

```
(%i3) invert(A);
```

$$(\%o3)\quad \begin{bmatrix} 1 & -\frac{3}{2} & \frac{1}{2} \\ -1 & 1 & 0 \\ -2 & 4 & -1 \end{bmatrix}$$

So $\begin{pmatrix} 2 & -1 & 1 \\ 2 & 0 & 1 \\ 4 & 2 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & -\frac{3}{2} & \frac{1}{2} \\ -1 & 1 & 0 \\ -2 & 4 & -1 \end{pmatrix}$.

(b) Again, first enter the matrix and calculate the determinant.

```
(%i4) B: matrix( [-1,0,3,-2], [4, 1/2, -1 ,0],
       [2,0,1,-1], [-1,2,1,-1] );
```

$$(\%o4)\quad \begin{bmatrix} -1 & 0 & 3 & -2 \\ 4 & \frac{1}{2} & -1 & 0 \\ 2 & 0 & 1 & -1 \\ -1 & 2 & 1 & -1 \end{bmatrix}$$

```
(%i5) determinant(B);
```

$$(\%o5)\quad -\frac{1}{2}$$

We have

$$\det\begin{pmatrix} -1 & 0 & 3 & -2 \\ 4 & \frac{1}{2} & -1 & 0 \\ 2 & 0 & 1 & -1 \\ -1 & 2 & 1 & -1 \end{pmatrix} = -\frac{1}{2},$$

so this matrix is invertible.

Now find the inverse.

```
(%i6) invert(B);
```

$$(\%o6)\quad \begin{bmatrix} -4 & -4 & 7 & 1 \\ -6 & -6 & 10 & 2 \\ -19 & -20 & 33 & 5 \\ -27 & -28 & 46 & 7 \end{bmatrix}$$

So $\begin{pmatrix} -1 & 0 & 3 & -2 \\ 4 & \frac{1}{2} & -1 & 0 \\ 2 & 0 & 1 & -1 \\ -1 & 2 & 1 & -1 \end{pmatrix}^{-1} = \begin{pmatrix} -4 & -4 & 7 & 1 \\ -6 & -6 & 10 & 2 \\ -19 & -20 & 33 & 5 \\ -27 & -28 & 46 & 7 \end{pmatrix}$.

Solution to Computer activity 56

(a)
```
(%i1) a[n]:=0.8^n;
```

$$(\%o1)\quad a_n := 0.8^n$$

(b)
```
(%i2) a[100];
```

$$(\%o2)\quad 2.037035976334508\,10^{-10}$$

(Remember, this means $2.037\,035\,976\,334\,508 \times 10^{-10}$.)

So $a_{100} = 2.037 \times 10^{-10}$ (to 4 s.f.).

Solution to Computer activity 57

(a)
```
(%i1) b[1]:1;
```

$$(\%o1)\quad 1$$

```
(%i2) b[n]:=2*b[n-1]+1;
```

$$(\%o2)\quad b_n := 2\,b_{n-1} + 1;$$

(b)
```
(%i3) b[5];
```

$$(\%o3)\quad 31$$

```
(%i4) b[100];
```

$$(\%o4)\quad 1267650600228229401496703205375$$

So $b_5 = 31$ and $b_{100} = 1\,267\,650\,600\,228\,229\,401\,496\,703\,205\,375$

(c)
```
(%i5) b[1]:10;
(%o5) 10
(%i6) b[5];
(%o6) 31
```

(d)
```
(%i7) kill(b);
(%o7) done
(%i8) b[1]:10;
(%o8) 10
(%i9) b[n]:=2*b[n-1]+1;
(%o9) b_n := 2 b_{n-1} + 1;
(%i10) b[5];
(%o10) 175
```

So b_5 is now 175.

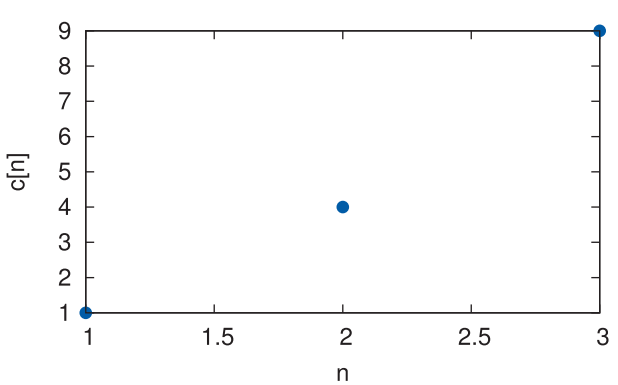
Solution to Computer activity 58

(a)
```
(%i1) c[n]:=n^2;
(%o1) c_n := n^2
```

(b)
```
(%i2) pts:[ [1,c[1]], [2,c[2]], [3,c[3]] ];
(%o2) [[1, 1], [2, 4], [3, 9]]
```

(c)
```
(%i3) wxplot2d([discrete, pts], [style, points],
 [xlabel, "n"], [ylabel, "c[n]"]);
```

(%t3)



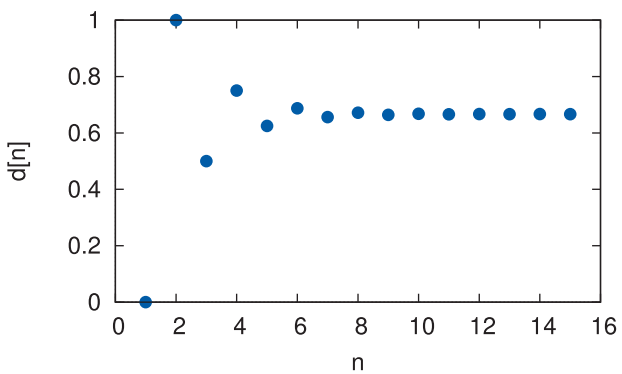
(%o3)

Solution to Computer activity 59

(a)
```
(%i1) d[1]:0;
(%o1) 0
(%i2) d[n]:=1/2*(2-d[n-1]);
(%o2) d_n := 1/2 (2 - d_{n-1})
```

(b)
```
(%i3) pts:makelist([n,d[n]], n, 1, 15);
```
(%o3) $[[1,0],[2,1],[3,\frac{1}{2}],[4,\frac{3}{4}],[5,\frac{5}{8}],[6,\frac{11}{16}],[7,\frac{21}{32}],[8,\frac{43}{64}],[9,\frac{85}{128}],$
$[10,\frac{171}{256}],[11,\frac{341}{512}],[12,\frac{683}{1024}],[13,\frac{1365}{2048}],[14,\frac{2731}{4096}],[15,\frac{5461}{8192}]]$

(c)
```
(%i4) wxplot2d([discrete, pts], [style, points],
       [xlabel, "n"], [ylabel, "d[n]"]);
```
(%t4)



(%o4)

Solution to Computer activity 60

(a) First, define the sequence.

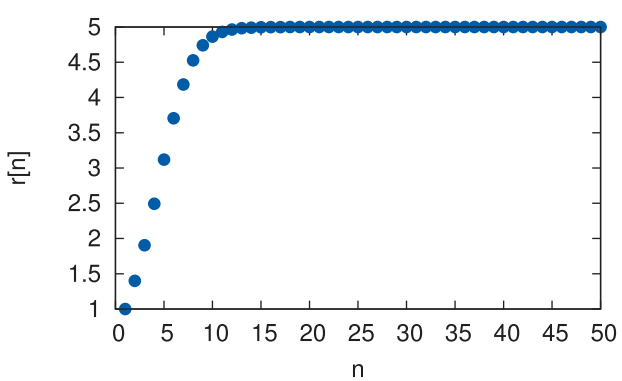
```
(%i1) r[1]:1;
```
(%o1) 1
```
(%i2) r[n]:=1.5*r[n-1]-0.1*(r[n-1])^2;
```
(%o2) $r_n := 1.5\,r_{n-1} - 0.1r_{n-1}^2$

Next, create a list of points to plot, suppressing the output.

```
(%i3) rpts:makelist([n,r[n]],n,1,50)$
```

Finally, plot the graph.

```
(%i4) wxplot2d([discrete, rpts], [style, points],
       [xlabel, "n"], [ylabel, "r[n]"]);
```
(%t4)



(%o4)

The terms of the sequence seem to be getting closer and closer to the value 5.

(b) First, define the sequence.

```
(%i5) s[n]:=n^3-10;
```

(%o5) $s_n := n^3 - 10$

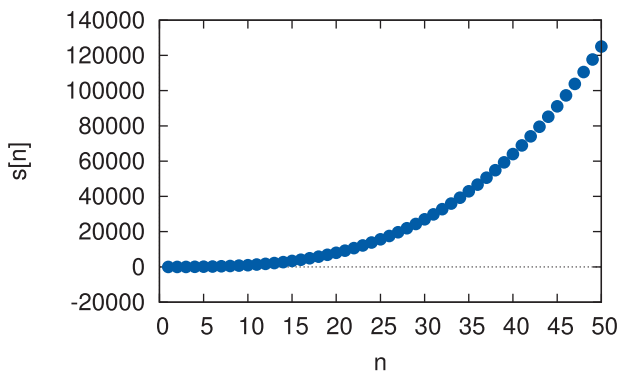
Next, create a list of 50 points to plot, suppressing the output.

```
(%i6) spts:makelist([n,s[n]],n,1,50)$
```

Finally, plot the graph.

```
(%i7) wxplot2d([discrete, spts], [style, points],
       [xlabel, "n"], [ylabel, "s[n]"]);
```

(%t7)



(%o7)

The terms of the sequence seem to be continually increasing.

(c) First, define the sequence.

```
(%i8) t[n]:=1-1.1^n;}
```

(%o8) $t_n := 1 - 1.1^n$

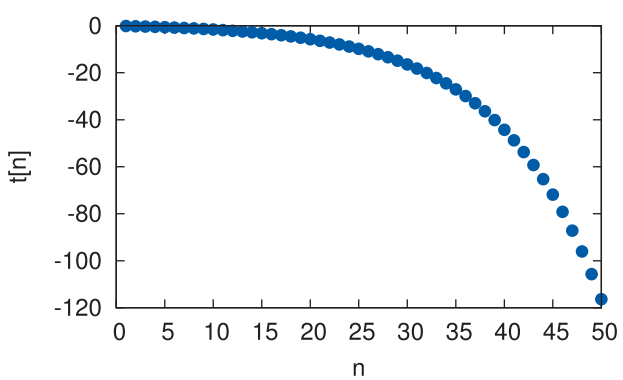
Next, create a list of 50 points to plot, suppressing the output.

```
(%i9) tpts:makelist([n,t[n]],n,1,50)$
```

Finally, plot the graph.

```
(%i10)wxplot2d([discrete, tpts], [style, points],
       [xlabel, "n"], [ylabel, "t[n]"]);
```

(%t10)



(%o10)

The terms of the sequence seem to be continually decreasing.

Solution to Computer activity 61

(a)
```
(%i1) sum(n^2, n, 1, 100);
```
```
(%o1) 338350
```
So $\sum_{n=1}^{100} n^2 = 338\,350$.

(b) The sum can be written $\sum_{n=1}^{20} \left(\frac{1}{3}\right)^n$ and so can be calculated as follows.
```
(%i2) sum((1/3)^n, n, 1, 20);
```
```
(%o2) 1743392200/3486784401
```
So $\sum_{n=1}^{20} \left(\frac{1}{3}\right)^n = \frac{1\,743\,392\,200}{3\,486\,784\,401}$.

Notice that the sum is a geometric series with first term $a = \frac{1}{3}$, common ratio $r = \frac{1}{3}$ and with 20 terms. So the sum can be calculated using

$$\begin{aligned}\frac{a(1-r^n)}{1-r} &= \frac{\frac{1}{3}\left(1-\left(\frac{1}{3}\right)^{20}\right)}{1-\frac{1}{3}}\\ &= \frac{\frac{1}{3}}{\frac{2}{3}}\left(1-\frac{1}{3^{20}}\right)\\ &= \frac{1}{2}\left(\frac{3^{20}-1}{3^{20}}\right)\\ &= \frac{3^{20}-1}{2\times 3^{20}}\\ &= \frac{3\,486\,784\,400}{2\times 3\,486\,784\,401}\\ &= \frac{1\,743\,392\,200}{3\,486\,784\,401}\\ &\approx 0.499\,999\,999\,8566\end{aligned}$$

(c)
```
(%i3) sum(1/(4^n), n, 1, inf);
```
```
(%o3) sum(1/4^n, n, 1, inf)
```

(d)
```
(%i4) simpsum:true;
```
```
(%o4) true
```
```
(%i5) sum(1/(4^n), n, 1, inf);
```
```
(%o5) 1/3
```
So $\sum_{n=1}^{\infty} \frac{1}{4^n} = \frac{1}{3}$. You could check this result using the expression for the sum of an infinite geometric series.

(e)
```
(%i6) sum(1/(n^2), n, 1, inf);
```

(%o6) $\dfrac{\pi^2}{6}$

So $\displaystyle\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$.

Solution to Computer activity 62

(a)
```
(%i1) taylor(cos(x), x, %pi/6, 3);
```

(%o1)/T/ $\dfrac{\sqrt{3}}{2} - \dfrac{x - \frac{\pi}{6}}{2} - \dfrac{\sqrt{3}\left(x - \frac{\pi}{6}\right)^2}{4} + \dfrac{\left(x - \frac{\pi}{6}\right)^3}{12} + \ldots$

So the cubic Taylor polynomial about $\pi/6$ for $f(x) = \cos(x)$ is

$$\frac{\sqrt{3}}{2} - \frac{x - \frac{\pi}{6}}{2} - \frac{\sqrt{3}\left(x - \frac{\pi}{6}\right)^2}{4} + \frac{\left(x - \frac{\pi}{6}\right)^3}{12}$$

(b)
```
(%i2) p(x)=''(taylor(tan(x), x, 0, 5));
```

(%o2)/T/ $p(x) := x + \dfrac{x^3}{3} + \dfrac{2x^5}{15} + \ldots$

So the quintic Taylor polynomial about 0 for $f(x) = \tan(x)$ is $x + \dfrac{x^3}{3} + \dfrac{2x^5}{15}$.

(c)
```
(%i3) p(y);
```

(%o3)/R/ $\dfrac{2y^5 + 5y^3 + 15y}{15}$

(d)
```
(%i4) float(p(0.1));
rat:  replaced 0.10033466666667 by 75251/750000 =
 0.10033466666667
(%o4)  0.10033466666667
```

So $p(0.1) = 0.100\,33$ (to 5 s.f.).

```
(%i5) float(tan(0.1)-p(0.1));
rat:  replaced 0.1003346666666667 by 75251/750000 =
 0.1003346666666667
rat:  replaced 0.1003346720854505 by 9334601/93034649 =
 0.1003346720854507
(%o5)  5.4187839916144221 10^-9
```

So the difference between the function and its quintic Taylor polynomial at $x = 0.1$ is 5.42×10^{-9} (to 3 s.f.).

Solution to Computer activity 63

First, define the function.

```
(%i1) f(x):=exp(sin(x^2));
```

(%o1) $f(x) := \exp(\sin(x^2))$

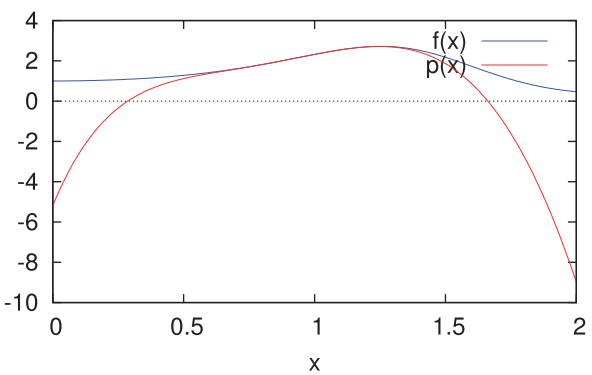
Next, calculate the required Taylor polynomial. The command is ended with `$` to suppress displaying the result.

```
(%i2) p(x):=''(taylor(f(x),x,1,5))$
```

Finally, plot the two curves over the required range.

```
(%i3) wxplot2d([f(x),p(x)], [x,0,2], [legend, "f(x)", "p(x)"]);
```

(%t3)



(%o3)

Solution to Computer activity 64

(a) `(%i1) a:3+4*%i;`

(%o1) $4\,\%i + 3$

(b) `(%i2) b:2-2*%i;`

(%o2) $2 - 2\,\%i$

(c) `(%i3) a+b;`

(%o3) $2\,\%i + 5$

`(%i4) a-b;`

(%o4) $6\,\%i + 1$

So `a+b`$= 5 + 2i$ and `a-b`$= 1 + 6i$.

`(%i5) a*b;`

(%o5) $(2 - 2\,\%i)(4\,\%i + 3)$

`(%i6) a/b;`

(%o6) $\dfrac{4\,\%i + 3}{2 - 2\,\%i}$

(d)
```
(%i7) rectform(a*b);
```
(%o7) $2\,\%i + 14$

```
(%i8) rectform(a/b);
```
(%o8) $\frac{7\,\%i}{4} - \frac{1}{4}$

So `a*b`$= 14 + 2i$ and `a/b`$= -\frac{1}{4} + \frac{7}{4}i$.

(e)
```
(%i9) realpart(a);
```
(%o9) 3

```
(%i10) imagpart(a);
```
(%o10) 4

So the real and imaginary parts of `a` are 3 and 4 as expected.

(f)
```
(%i11) conjugate(a);
```
(%o11) $3 - 4\,\%i$

So the conjugate of `a` is $3 - 4i$ as expected.

(g)
```
(%i12) abs(b);
```
(%o12) $2^{3/2}$

```
(%i13) carg(b);
```
(%o13) $-\frac{\pi}{4}$

So $|$`b`$| = 2^{3/2} = \left(\sqrt{2}\right)^3$ and the principal argument of `b` is $-\frac{\pi}{4}$.

Solution to Computer activity 65

(a)
```
(%i1) c:3*(cos(5*%pi/13)+%i*sin(5*%pi/13));
```
(%o1) $3\left(\%i\sin\left(\frac{5\pi}{13}\right) + \cos\left(\frac{5\pi}{13}\right)\right)$

```
(%i2) d:4*(cos(7*%pi/13)+%i*sin(7*%pi/13));
```
(%o2) $4\left(\%i\sin\left(\frac{7\pi}{13}\right) + \cos\left(\frac{7\pi}{13}\right)\right)$

(b) The modulus of `c*d` can be found as follows.

```
(%i3) abs(c*d);
```
(%o3) $12\sqrt{\sin\left(\frac{5\pi}{13}\right)^2 + \cos\left(\frac{5\pi}{13}\right)^2}\sqrt{\sin\left(\frac{7\pi}{13}\right)^2 + \cos\left(\frac{7\pi}{13}\right)^2}$

Simplify this using the `trigreduce` command. Remember, `%` refers to the result of the previous calculation.

```
(%i4) trigreduce(%);
```
(%o4) 12

The principal argument of `c*d` can be found as follows. Again the initial result requires simplification.

```
(%i5) carg(c*d);
```

$$\texttt{(\%o5)}\quad \texttt{atan}\left(\frac{\texttt{sin}\left(\frac{7\pi}{13}\right)}{\texttt{cos}\left(\frac{7\pi}{13}\right)}\right)+\texttt{atan}\left(\frac{\texttt{sin}\left(\frac{5\pi}{13}\right)}{\texttt{cos}\left(\frac{5\pi}{13}\right)}\right)+\pi$$

```
(%i6) trigreduce(%);
```

$$\texttt{(\%o6)}\quad \frac{12\pi}{13}$$

So $|\texttt{c*d}| = 12$ and $\arg(\texttt{c*d}) = \dfrac{12\pi}{13}$.

Similar methods can be used to find the modulus and principal argument of `c/d`.

```
(%i7) abs(c/d);
```

$$\texttt{(\%o7)}\quad \frac{3\sqrt{\texttt{sin}\left(\frac{5\pi}{13}\right)^2+\texttt{cos}\left(\frac{5\pi}{13}\right)^2}}{4\sqrt{\texttt{sin}\left(\frac{7\pi}{13}\right)^2+\texttt{cos}\left(\frac{7\pi}{13}\right)^2}}$$

```
(%i8) trigreduce(%);
```

$$\texttt{(\%o8)}\quad \frac{3}{4}$$

```
(%i9) carg(c/d);
```

$$\texttt{(\%o9)}\quad -\texttt{atan}\left(\frac{\texttt{sin}\left(\frac{7\pi}{13}\right)}{\texttt{cos}\left(\frac{7\pi}{13}\right)}\right)+\texttt{atan}\left(\frac{\texttt{sin}\left(\frac{5\pi}{13}\right)}{\texttt{cos}\left(\frac{5\pi}{13}\right)}\right)+\pi$$

```
(%i10) trigreduce(%);
```

$$\texttt{(\%o10)}\quad \frac{24\pi}{13}$$

Note that although `carg` gives a principal argument, that is an argument in the interval $(-\pi, \pi]$, the `trigreduce` command simplifies the expression to an equivalent argument which is outside this interval.

The principal argument is $\dfrac{24\pi}{13} - 2\pi = -\dfrac{2\pi}{13}$.

So $|\texttt{c/d}| = \frac{3}{4}$ and $\arg(\texttt{c/d}) = -\dfrac{2\pi}{13}$.

Solution to Computer activity 66

(a) `(%i1) solns:solve(z^3+z+1=0);`

$$(\%o1)\quad [z = \left(\frac{\sqrt{31}}{2\,3^{3/2}} - \frac{1}{2}\right)^{1/3}\left(-\frac{\sqrt{3}\,\%i}{2} - \frac{1}{2}\right) - \frac{\frac{\sqrt{3}\,\%i}{2} - \frac{1}{2}}{3\left(\frac{\sqrt{31}}{2\,3^{3/2}} - \frac{1}{2}\right)^{1/3}},$$

$$z = \left(\frac{\sqrt{31}}{2\,3^{3/2}} - \frac{1}{2}\right)^{1/3}\left(\frac{\sqrt{3}\,\%i}{2} - \frac{1}{2}\right) - \frac{-\frac{\sqrt{3}\,\%i}{2} - \frac{1}{2}}{3\left(\frac{\sqrt{31}}{2\,3^{3/2}} - \frac{1}{2}\right)^{1/3}},$$

$$z = \left(\frac{\sqrt{31}}{2\,3^{3/2}} - \frac{1}{2}\right)^{1/3} - \frac{1}{3\left(\frac{\sqrt{31}}{2\,3^{3/2}} - \frac{1}{2}\right)^{1/3}}]$$

(b) `(%i2) float(rectform(solns));`

$$(\%o2)\quad [z = 0.34116390191401 - 1.161541399997252\,\%i,$$
$$z = 1.161541399997252\,\%i + 0.34116390191401,$$
$$z = -0.68232780382802]$$

So, to two decimal places, the solutions are

$$z = 0.34 - 1.16i, \quad z = 0.34 + 1.16i, \quad \text{and} \quad z = -0.68.$$

(c) `(%i3) solve(z^4-20*z^3+171*z^2-626*z+962=0);`

$$(\%o3)\quad [z = 3 - 2\,\%i, z = 2\,\%i + 3, z = 7 - 5\,\%i, z = 5\,\%i + 7]$$

So the solutions are $3 - 2i$, $3 + 2i$, $7 - 5i$ and $7 + 5i$.

Solution to Computer activity 67

(a) `(%i1) s:solve(z^4+4*z+5=0);`

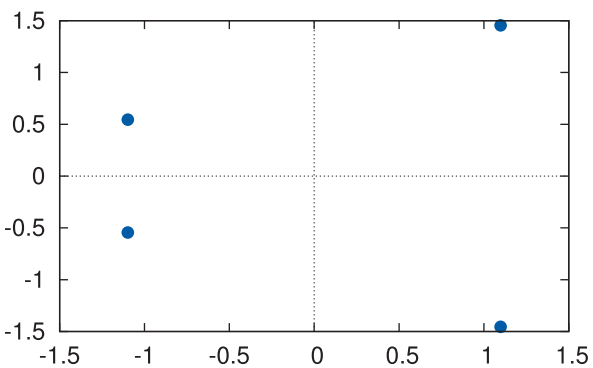
$$(\%o1)\quad [z = -\%i - \frac{\sqrt{4 - 4\,\%i}}{2}, z = \frac{\sqrt{4 - 4\,\%i}}{2} - \%i,$$
$$z = \%i - \frac{\sqrt{4\,\%i + 4}}{2}, z = \frac{\sqrt{4\,\%i + 4}}{2} + \%i]$$

(b) `(%i2) v:makelist(rhs(s[k]), k, 1, length(s));`

$$(\%o2)\quad [-\%i - \frac{\sqrt{4 - 4\,\%i}}{2}, \frac{\sqrt{4 - 4\,\%i}}{2} - \%i,$$
$$\%i - \frac{\sqrt{4\,\%i + 4}}{2}, \frac{\sqrt{4\,\%i + 4}}{2} + \%i]$$

(c)
```
(%i3) pts:makelist([realpart(v[k]), imagpart(v[k])],
       k, 1, length(s));
```
(%o3) $[[-\frac{\sqrt{2^{5/2}+4}}{2^{3/2}}, \frac{\sqrt{2^{5/2}-4}}{2^{3/2}} - 1], [\frac{\sqrt{2^{5/2}+4}}{2^{3/2}}, -\frac{\sqrt{2^{5/2}-4}}{2^{3/2}} - 1],$
$[-\frac{\sqrt{2^{5/2}+4}}{2^{3/2}}, 1 - \frac{\sqrt{2^{5/2}-4}}{2^{3/3}}], [\frac{\sqrt{2^{5/2}+4}}{2^{3/2}}, \frac{\sqrt{2^{5/2}-4}}{2^{3/2}} + 1]]$

(d)
```
(%i4) wxplot2d([discrete,pts], [style,points],
       [xlabel,""], [ylabel,""]);
```
(%t4)



(%o4)

Solution to Computer activity 68

```
(%i1) s:solve(z^7=1);
```
(%o1) $[z = \%e^{\frac{2\%i\pi}{7}}, z = \%e^{\frac{4\%i\pi}{7}}, z = \%e^{\frac{6\%i\pi}{7}},$
$z = \%e^{-\frac{6\%i\pi}{7}}, z = \%e^{-\frac{4\%i\pi}{7}}, z = \%e^{-\frac{2\%i\pi}{7}},$
$z = 1]$

The solutions are given in terms of exponential functions. You will learn about the exponential form of complex numbers in Section 4 of Unit 12.

The solutions can be written in Cartesian form as follows.

```
(%i2) float(rectform(s));
```
(%o2) $[z = 0.78183148246803\,\%i + 0.62348980185873,$
$z = 0.97492791218182\,\%i - 0.22252093395631,$
$z = 0.43388373911756\,\%i - 0.90096886790242,$
$z = -0.43388373911756\,\%i - 0.90096886790242,$
$z = -0.97492791218182\,\%i - 0.22252093395631,$
$z = 0.62348980185873 - 0.78183148246803\,\%i,$
$z = 1.0]$

So, to two decimal places, the seven seventh roots of unity are
$z = 0.62 + 0.78i$, $z = -0.22 + 0.97i$, $z = -0.90 + 0.43i$,
$z = -0.90 - 0.43i$, $z = -0.22 - 0.97i$, $z = 0.62 - 0.78i$
and $z = 1.00$.

To plot the solutions in the complex plane, first create a list containing just the solutions of the equation (without the z=). The command is ended with $ to suppress the output.

```
(%i3) v:makelist(rhs(s[k]), k, 1, length(s))$
```

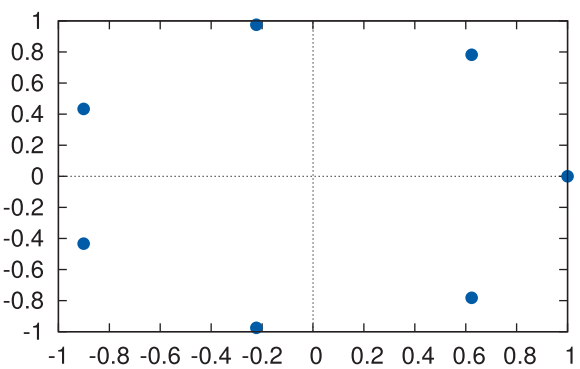
Then create the list of points to be plotted.

```
(%i4) pts:makelist([realpart(v[k]), imagpart(v[k])],
       k, 1, length(s))$
```

Finally, plot the points.

```
(%i5) wxplot2d([discrete,pts], [style,points],
       [xlabel,""], [ylabel,""]);
```

(%t5)



(%o5)

Notice that the complex solutions occur in conjugate pairs, and hence lie symmetrically about the horizontal axis. The solutions also lie on the unit circle (which does not seem circular here due to the different axis scales used).

Maxima reference guide

Mathematical operations and functions

Operation or function	Syntax	Example
Addition	`+`	`4+3;`
Subtraction	`-`	`4-3;`
Multiplication	`*`	`4*3;`
Division	`/`	`4/3;`
Brackets	`(` *and* `)`	`2*(3+4);`
Powers, for example 2^3	`^`	`2^3;`
	or `**`	`2**3;`
Square root, for example $\sqrt{5}$	`sqrt(` `)`	`sqrt(5);`
Exponential, for example e^3	`%e^`	`%e^3;`
	or `exp(` `)`	`exp(3);`
Natural logarithm, ln	`log(` `)`	`log(8);`
Magnitude (absolute value) of a real number, or modulus of a complex number	`abs(` `)`	`abs(-3);` `abs(1+2*%i);`
sin	`sin(` `)`	`sin(1);`
cos	`cos(` `)`	`cos(3*%pi/2);`
tan	`tan(` `)`	`tan(%pi/4);`
cosec	`csc(` `)`	`csc(2);`
sec	`sec(` `)`	`sec(%pi/3);`
cot	`cot(` `)`	`cot(%pi/4);`
$\sin^{-1}$	`asin(` `)`	`asin(sqrt(3)/2);`
$\cos^{-1}$	`acos(` `)`	`acos(1/sqrt(2));`
$\tan^{-1}$	`atan(` `)`	`atan(1/2);`
Equality (within an equation)	`=`	`5=2*x+3;`
Matrix addition	`+`	`P+Q;`
Matrix subtraction	`-`	`P-Q;`
Scalar multiple of a matrix	`*`	`2*P;`
Matrix multiplication	`.`	`P.Q;`
Matrix powers	`^^`	`P^^3;`
Determinant (of a square matrix)	`determinant(` `matrix` `)`	`determinant(P);`
Matrix inverse (of a square matrix)	`invert(` `matrix` `)`	`invert(P);`
	or `matrix` `^^(-1)`	`P^^(-1);`
Identity matrix	`ident(` `size` `)`	`ident(2);`

Mathematical operations and functions (continued)

Operation or function	Syntax	Example
Real part of a complex number	realpart( )	realpart(1+2*%i);
Imaginary part of a complex number	imagpart( )	imagpart(1+2*%i);
Conjugate of a complex number	conjugate( )	conjugate(1+2*%i);
Principal argument of a complex number	carg( )	carg(1+2*%i);
Express a complex number in Cartesian form	rectform( )	rectform(2/%i);

Mathematical commands

Operation	Command	Example
Get help on a command	? command *or* describe(command)	? float; describe(float);
Find help information whose title contains the given text	?? *or* describe(, inexact)	?? flo; describe(flo, inexact);
Force the evaluation of a command	''( )	g(x):=''(diff(x^4,x));
Load a package	load(package name)	load(implicit_plot);
Assign a value to a variable	variable : value	a:23;
Display the value of a variable	variable	a;
Define a function	:=	f(x):=2*x+3;
Evaluate a function at a value	function( )	f(1);
Remove an assigned variable or function	kill(variable) kill(function)	kill(a); kill(f);
Remove all assigned variables and functions	kill(all)	kill(all);
Reset all system variables	reset()	reset();
Convert to a decimal number	float( )	float(sqrt(2));
Expand brackets	expand( )	expand((x+1)^2);
Factorise	factor( )	factor(2*x+4*x^2);
Simplify	fullratsimp( )	fullratsimp((2*x+4*x^2)/x);
Simplify something involving exponentials and logarithms	radcan( )	radcan(log(x^2*y));
Combine logarithms	logcontract( )	logcontract(log(a)+log(b));

Mathematical commands (continued)

Operation	Command	Example
Expand trigonometric functions of sums, differences and multiples of angles	`trigexpand( )`	`trigexpand(sin(A+B));`
Express powers of sines and cosines in terms of sines and cosines of multiple angles	`trigreduce( )`	`trigreduce(sin(x)^2);`
Simplify trigonometric expressions	`trigsimp( )`	`trigsimp(sin(x)^2+cos(x)^2);`
Simplify algebraic fractions containing trigonometric functions	`trigrat( )`	`trigrat((sin(2*x))/cos(x));`
Substitute	`subst(value,variable,expression)`	`subst(4,x,x^2+1);` which substitutes 4 for x in x^2+1
Left-hand side of an equation	`lhs(equation)`	`lhs(4*x+1=2*x-2);`
Right-hand side of an equation	`rhs(equation)`	`rhs(4*x+1=2*x-2);`
Solve an equation exactly	`solve(equation)`	`solve(2*x^2-1=0);`
Solve an equation for a variable, i.e. change the subject	`solve(equation, variable)`	`solve(2*a*b-3*b=0, a);`
Solve simultaneous equations	`solve(list of equations, list of variables)`	`solve([2*x+y=4, x-2*y=1], [x,y]);`
Solve an equation numerically	`find_root(expression,variable, interval start value, interval end value)`	`find_root(2*x^3-1,x,0,1);`
Form a list	`[ , ,...]`	`[a,b,c];`
Assign a list to a variable	`variable:[ , ,...]`	`A:[a,b,c];`
Create a list	`makelist(general term, dummy variable,start value, end value)`	`makelist(2*n,n,1,100);`
Extract an element of a list	`list[index]`	`A[2];`
Find the length of a list	`length(list)`	`length(A);`
Specify a matrix	`matrix(row,row,...)`	`matrix([1,2],[3,4]);`
Size of a matrix	`matrix_size(matrix)`	`matrix_size(A);`
Row of a matrix	`matrix[row]`	`A[2];`
Element of a matrix	`matrix[row,column]` *or* `matrix[row][column]`	`A[3,4];` `A[3][4];`

Mathematical commands (continued)

Operation	Command	Example
Define a sequence using a closed form	`sequence[n]:=expression in n`	`a[n]:=n^2;`
Define a sequence using a using a recurrence system	`sequence[first term number]: initial value` `sequence[n]:= expression in sequence[n-1]`	`b[1]:1;` `b[n]:=2*b[n-1];`
Calculate a term in a sequence	`sequence[term number]`	`b[50];`
Sum a series	`sum(general term,index variable, lower limit,upper limit)`	`sum(n^2,n,1,10);`
Differentiate	`diff( ,variable)`	`diff(sin(x^2),x);`
Differentiate multiple times	`diff( ,variable,positive integer)`	`diff(log(x),x,3);` to differentiate three times
Integrate (find an antiderivative)	`integrate( ,variable)`	`integrate(x^2,x);`
Evaluate a definite integral	`integrate( ,variable, lower limit,upper limit)`	`integrate(sin(x),x,0,1);`
Find an approximate value of a definite integral	`quad_qags( ,variable, lower limit,upper limit)`	`quad_qags(exp(-x^2),x,0,1);`
Find a Taylor polynomial	`taylor(function,variable, centre,degree)`	`taylor(sin(x),x,0,3);`
Make an assumption about a variable	`assume( )`	`assume(a>0);`
State two things are equal	`equal( , )`	`assume(equal(n,3));`
State two things are not equal	`notequal( , )`	`assume(notequal(n,-1));`
Forget a property	`forget(property)`	`forget(a>0);`
List all known facts	`facts()`	`facts();`
List all known facts about a particular variable	`facts(variable)`	`facts(a);`

Mathematical commands (continued)

Operation	Command	Example
Plot a graph	`wxplot2d(` *expression* `,` *horizontal range* `,...)`	`wxplot2d(x^2,[x,0,1]);` which plots the graph of x^2 for x between 0 and 1 `wxplot2d(x^2,[x,0,1],[y,0,2]);` which plots the graph of x^2 for x between 0 and 1, and with vertical axis between 0 and 2
Plot several graphs	`wxplot2d(` *list of expressions* `,` *horizontal range* `,...)`	`wxplot2d([x^2,2*x],[x,0,1]);` which plots graphs of x^2 and $2x$ for x between 0 and 1
Plot a sequence	`wxplot2d([discrete,` *list of coordinate pairs* `],` `[style, points],...)` (use `plot2d` similarly if not using wxMaxima)	`wxplot2d([discrete, [[1,5], [2,10]]], [style, points]);`
Plot a curve represented by an equation in implicit form	`wximplicit_plot(` *equation* `,` *horizontal range* `,` *vertical range* `, ...)`	`wximplicit_plot(x^2+2*y^2=1, [x,-1,1], [y,-1,1]);`
Plot several curves represented by equations in implicit form	`wximplicit_plot(` *list of equations* `,` *horizontal range* `,` *vertical range* `, ...)` (use `implicit_plot` similarly if not using wxMaxima) *Note*: the `wximplicit_plot` and `implicit_plot` commands require the `implicit_plot` package to be loaded.	`wximplicit_plot( [x^2+2*y^2=1, y=x^3], [x,-1,1], [y,-1,1]);`
Set the default plotting options	`set_plot_option(` *option* `)`	`set_plot_option(same_xy);`

Other mathematical symbols

Constant	Syntax	Example
e	%e	%e^2;
π	%pi	2*%pi;
i	%i	2+3*%i;
∞	inf	sum(1/n^2,n,1,inf);

Maxima system variables

Variable	Description	Example
fpprintprec	The number of significant figures of a decimal number to display	fpprintprec:3;
functions	The list of all user-defined functions	functions;
values	The list of all user-assigned variables	values;
simpsum	Whether to simplify summations	simpsum:true;

Maxima graph plotting options

The following options can be added as additional arguments to plotting commands.

Option	Argument	Example
Vertical range	[y, ,]	[y, 0, 5]
Curve colour	[color, , ...]	[color, red]
Legend text	[legend, " ", ...]	[legend, "Car A"]
Turn legend off	[legend, false]	[legend, false]
Position legend at the bottom of the graph	[gnuplot_preamble, "set key bottom"]	[gnuplot_preamble, "set key bottom"]
Position legend to the right of the graph	[gnuplot_preamble, "set key outside"]	[gnuplot_preamble, "set key outside"]
Horizontal axis label	[xlabel, " "]	[xlabel, "t (h)"]
Vertical axis label	[ylabel, " "]	[ylabel, "s (km)"]
Set axes to have equal scales	[same_xy, true]	[same_xy, true]
Change the size of the graph	wxplot_size:[,]	wxplot_size:[800,600];
Plot points	[style, points]	[style, points]

Maxima packages

Maxima package	Functions added
implicit_plot	Plotting curves given by equations in implicit form

wxMaxima keyboard sequences (Microsoft Windows)

Function	Key sequence
Zoom in	Alt-I
Zoom out	Alt-O
Copy	Ctrl-C
Interrupt a calculation	Ctrl-G
Re-evaluate the worksheet	Ctrl-R
Close wxMaxima	Ctrl-Q
Paste	Ctrl-V

wxMaxima menu commands (Microsoft Windows)

Operation	Menu name	Menu option
Close wxMaxima	File	Exit
Save your work	File	Save or Save As
Print the worksheet	File	Print
Open a worksheet	File	Open
Configure wxMaxima	Edit	Configure
Copy	Edit	Copy
Paste	Edit	Paste
Zoom in	View	Zoom In
Zoom out	View	Zoom Out
Toggle display of main toolbar	View	Main Toolbar
Re-evaluate worksheet	Cell	Evaluate All Visible Cells
Insert text cell	Cell	Insert Text cell
Insert title cell	Cell	Insert Title cell
Insert section cell	Cell	Insert Section cell
Insert subsection cell	Cell	Insert Subsection cell
Interrupt a calculation	Maxima	Interrupt
Restart Maxima	Maxima	Restart Maxima
Enter a matrix	Algebra	Enter Matrix...
Maxima help window	Help	Maxima Help

Maxima command-line interface commands

Operation	Command
Close Maxima	`quit();`
Interrupt a calculation	Ctrl-C
Turn off 2D formatting of maths	`display2d:false;`
Display output left justified	`leftjust:true;`
Start writing a transcript	`writefile("file path");`
Stop writing a transcript	`closefile();`
Playback all the input and output in a session	`playback();`
Playback a range of input and output	`playback([start line number, end line number]);`
Save the state of a session	`save("file path", all);`
Reload a session	`load("file path");`

Acknowledgements

Grateful acknowledgement is made to the following sources:

Page 5: Taken from: http://www.ma.utexas.edu/people/in_memoriam/ (Accessed 2014)

Page 6: www.cartoonstock.com

Every effort has been made to contact copyright holders. If any have been inadvertently overlooked the publishers will be pleased to make the necessary arrangements at the first opportunity.

Index

abort calculation
  using the command-line interface 95
  using wxMaxima 19
`abs` 14
  for complex numbers 85
absolute value 14
accessibility 89
  changing wxMaxima colours and fonts 90
  using a screen magnifier 91
  using a screenreader 91
  using wxMaxima with keyboard alone 90
`acos` 47
adding text to a wxMaxima worksheet 26
addition 14
  of matrices 71
annotating a wxMaxima worksheet 26
approximation, numerical 50
argument of a command 14
argument of a complex number 85
`asin` 47
assign
  a derivative to function 60
  an integral to function 63
  functions 42
  variable 22
`assume` 65, 66
assumption
  about a variable 65, 66
  equals 66
  forget 65, 66
  list all 66
  not equals 65, 66
`atan` 47
axis scales, on a graph 55

brackets 14

`carg` 85
Cartesian form of a complex number 85
CAS *see* computer algebra system
cell, in wxMaxima 16
  compress 16
  delete 17
  marker 11, 16
  re-evaluate 20
  section 26
  subsection 26
  text 26
  title 26
changing a sequence 77
changing the size of a graph, in wxMaxima 41
circles, plotting 54
closed form of a sequence 75
`closefile` 98
closing
  command-line interface 94
  wxMaxima 13
colour of graph 40
command
  argument of 14
  editing
    using the command-line interface 95
    using wxMaxima 20
  force evaluation of 61
  reusing
    using the command-line interface 95
    using wxMaxima 20
command-line interface 7, 91
  abort calculation 95
  closing 94
  creating transcript 97
  cursor, editing 92
  editing commands 95
  editing cursor 92
  graphs 100
  input line number 92
  interrupt calculation 95
  output line number 93
  plotting graphs 100
  printing 97
  reusing commands 95
  saving work 96
  transcript 97
complex numbers 35, 84
  argument (principal) 85
  Cartesian form 85
  conjugate 85
  imaginary part 85
  modulus 85
  polar form 85
  real part 85
  rectangular form 85

compress cell, in wxMaxima 16
computer algebra system 5
configure wxMaxima 12, 16
`conjugate` 85
conjugate of a complex number 85
copy, using wxMaxima 21
`cos` 47
cosec 47
`cot` 47
create a list using `makelist` 80
`csc` 47
cube roots 15
cursor
  editing
    in command-line interface 92
    in wxMaxima 20
  horizontal, in wxMaxima 20

decimal approximation of a number 17
define
  functions 42
  variables 22
definite integral 62
  approximate value 66
delete
  cell, in wxMaxima 17
  function 43
  variable 24
derivative 59
  assign to function 61
  second, third 60
`describe` 31
`determinant` 74
determinant of a matrix 74
`diff` 59
differentiate 59
division 14

editing commands
  using the command-line interface 95
  using wxMaxima 20
editing cursor
  using the command-line interface 92
  using wxMaxima 20
element
  index 36
  of a list 36
  of a matrix 71

`equal` 66
equals 34
  assumption 66
equations 34
  changing the subject 37
  left-hand side 34
  manipulating 37
  plotting 39
  rearranging 37
  right-hand side 34
  simultaneous 38
  solving 34
  solving numerically 50
  trigonometric 49
`erf` 64
evaluate
  an expression 11
  an expression using the command-line interface 93
  force command to 61, 62
  re-evaluate a cell using wxMaxima 20
  re-evaluate worksheet using wxMaxima 22
`exp` 46
`expand` 33
expanding brackets 33
exponentials 44
expressions, evaluating 11

`factor` 33
factorising 33
facts *see* assumption
`facts` 66
file types, wxMaxima 27
`find_root` 50
`float` 17
floating-point number 17
  displayed precision 25
`forget` 65, 66
forget assumption 65, 66
`fpprintprec` 25
front-ends 7
`fullratsimp` 33
functions 42
  assign 42
  define 42
  delete 43
  trigonometric 47
  trigonometric, inverse 47

`functions` 43

`gamma_incomplete` 64
graphs
  changing default behaviour 56
  changing line thickness 100
  changing properties 100
  colour 40
  equal scales 55
  from equations in implicit form 54
  legend 40
  plotting 39
  size of, in wxMaxima 41
  using command-line interface 100
  vertical range 39

help 30
horizontal cursor, in wxMaxima 20

`ident` 75
identity matrix 75
imaginary numbers 35
imaginary part of a complex number 85
`imagpart` 85
implicit form of an equation, plotting 54
`implicit_plot`
  command 54
  package 54
indefinite integral 62
index (of a list element) 36
`inf` 82
infinite sequence 80
infinity 82
input line number 11
input prompt, wxMaxima 11
inputting a matrix 68
installing Maxima 8
integral 62
  assign to a function 63
  definite 62
  definite, approximate value 66
  indefinite 62
integrate 62
`integrate` 62
interfaces 7
  command-line 7, 91
  wxMaxima 7, 9
interrupt calculation
  using the command-line interface 95
  using wxMaxima 19
inverse of a matrix 74
inverse trigonometric functions 47
`invert` 74

keyword 40
`kill`
  function 43
  variable 24

left-hand side (of an equation) 34
legend, of a graph 40
`length` 88
length of a list 88
`lhs` 34
line number
  input 11
  output 11
list 36
  assumptions 66
  create using `makelist` 80
  element of 36
  index 36
  length of 88
`load` 99
  package 54
  previous session
    using the command-line interface 99
    using wxMaxima 28
loading packages 54
`log` 46
logarithms 44

magnitude 14
`makelist` 80
manipulating
  equations 37
  trigonometric expressions 52
mathematical operations 14
matrix 68
  addition 71
  determinant 74
  element 71
  identity 75
  inputting 68
  inverse 74
  multiplicaton 71
  powers 71

row 71
scalar multiplication 71
size 71
subtraction 71
matrix 68
matrix_size 71
Maxima 5
installing 8
menus, wxMaxima 9
modulus of a complex number 85
multiplication 14
matrix 71
scalar and matrix 71

not equals, assumption 65, 66
notequal 65, 66
numbers, complex 84
numerical approximation to a solution 50
numerical solution 50

output line number 11

packages 54
implicit_plot 54
loading 54
paste, using wxMaxima 21
playback 98
plot2d 39
using command-line interface 100
plotting
changing default behaviour 56
changing line thickness 100
changing properties 100
circles 54
colour 40
complex numbers 87
equal scales 55
equations in implicit form 54
graphs 39
points 78
sequences 78
size of graph, in wxMaxima 41
using command-line interface 100
point, plotting 78
polar form of a complex number 85
polynomial, Taylor 82
powers 14
matrix 71
principal argument of a complex number 85
printing
using the command-line interface 97
using wxMaxima 29
prompt, wxMaxima 11

quad_qags 67
quadrature 67

re-evaluate
a cell using wxMaxima 20
a worksheet using wxMaxima 22
real part of a complex number 85
realpart 85
rearranging an equation 37
rectangular form of a complex number 85
rectform 85
recurrence relation for a sequence 76
reset 25
resetting system variables 25
reusing commands
using the command-line interface 95
using wxMaxima 20
revising a sequence 77
rhs 34
right-hand side (of an equation) 34
roots
cube 15
square 14
row of a matrix 71

save 99
saving your work
using the command-line interface 96
using wxMaxima 27
scalar multiplication, matrix 71
scales, on graph 55
Schelter, William 5
scientific notation 18
screen magnifier, use with wxMaxima 91
screenreader 91
screenshot 29
sec 47
section cell, wxMaxima 26
sequences 75
closed form 75
infinite 80
plotting 78

recurrence relation 76
revising 77
term of 77
series
summing 81
Taylor 82
`set_plot_option` 56
significant figures 25
simplifying
expressions 33
trigonometric expressions 52
simultaneous equations 38
`sin` 47
size of a matrix 71
size of graph, in wxMaxima 41
`solve` 34, 86
solving equations 34
trigonometric 49
numerically 50
with non-real solutions 86
`sqrt` 14
square root 14
subject of equation, changing 37
subsection cell, wxMaxima 26
`subst` 33
substitute 33
subtraction 14
of matrices 71
`sum` 81
summing series 81
symbolic computation 17
syntax 14
system variables 25
resetting 25

table of contents, wxMaxima 10
`tan` 47
`taylor` 84
Taylor polynomials 82
term of a sequence 77
text cell, wxMaxima 26
title cell, wxMaxima 26
toolbar, wxMaxima 9
transcript, creating using the command-line interface 97
`trigexpand` 53
trigonometric equations 49
trigonometric expression
manipulating 52
simplifying 52
trigonometric functions 47
inverse 47
`trigrat` 53
`trigreduce` 52
`trigsimp` 53

variable 22
assign 22
assumption 65, 66
define 22
delete 24
system 25

worksheet
wxMaxima 9
annotate 26
re-evaluate 22
`writefile` 97
`wximplicit_plot` 54
wxMaxima 7, 9
abort calculation 19
adding text 26
annotating 26
cell 16
compress 16
delete 17
marker 11, 16
re-evaluate 20
section 26
subsection 26
text 26
title 26
changing colours and fonts 90
closing 13
compress cell 16
configure 12, 16
copy 21
cursor
editing 20
horizontal 20
delete cell 17
editing commands 20
editing cursor 20
file types 27
horizontal cursor 20

input line number 11
input prompt 11
interrupt calculation 19
line number
input 11
output 11
menus 9
output line number 11
paste 21
printing 29
prompt, input 11
re-evaluate
cell 20
worksheet 22
reusing commands 20
saving work 27
table of contents 10
toolbar 9
use with keyboard alone 90
worksheet 9
re-evaluate 22
`wxplot2d` 39
discrete points 79
`wxplot_size` 41